

Scaled Agile Framework

SAFe 6.0

Select SAFe configuration

OVERVIEW ESSENTIAL LARGE SOLUTION PORTFOLIO FULL

[Adjust Cookie Settings](#)

Extended SAFe Guidance

This guidance extends the body of knowledge

Recent Blog Posts

Below is a list of the most

beyond the articles that are linked directly from the Big Picture. Some highlights below.

- [What's New in SAFe 6.0](#)
- [Accelerating Flow in SAFe](#)
- [Agile Contracts](#)
- [Agile Marketing](#)
- [Applying Kanban in SAFe](#)
- [Applying SAFe to Hardware Development](#)
- [Agile Architecture](#)

recent blog posts about SAFe.

- [Say Hello to SAFe 6.0!](#)
- [Framework Promotions](#)
- [Exciting new book: SAFe for DevOps Practitioners](#)

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



The cloud is our future. We have fundamentally no choice.

—Satya Nadella, CEO
Microsoft

The Cloud and SAFe

The Cloud represents virtual, on-demand processing and storage services used for cost-effective and scalable infrastructure and operations, implementation of the Adjust CooDevOps toolchain, and development and hosting of AI applications.

The ever-expanding universe of cloud capabilities has fundamentally changed how digitally enabled solutions are built, deployed, and maintained. In fact, cloud computing is the single most disruptive driver of delivery model change that enterprise IT has faced since its inception [1]. Not surprisingly, the number one reason enterprises are moving to the cloud so quickly is to increase product development speed and agility [2].

The cloud is everywhere, and it fuels digital business. SAFe enterprises can harness the power and ubiquity of the cloud to increase agility in all areas of the organization.

Details

As the cloud has evolved from a technology disruptor to a business disruptor [3], it now permeates all facets of the enterprise. It supplies computing resources on-demand to IT operations and engineering teams. It provides professionals across business functions with fast, reliable access to applications that boost collaboration and productivity. It enables artificial intelligence (AI) and machine learning (ML) capabilities that allow enterprises to achieve major advancements in product innovation.

Digitally savvy enterprises leverage cloud capabilities to transform how their technology is managed and how they deliver new products to market. The result is an enterprise that responds quickly to dynamic market forces and engineering teams that serve as value engines for the business. This is achieved by combining cloud capabilities with Lean-Agile practices in key [Development Value Streams](#) (DVS).

Aligning the Cloud to Value

When the cloud hosts enterprise systems, those systems run more reliably and are maintained more efficiently. This transforms the technology landscape and can reduce operating costs. But innovation is accelerated when cloud capabilities are also used to enable the DVS (Figure 1). This transforms the business landscape and can generate new value quickly and continuously for the enterprise.

Figure 1. The cloud accelerates the development value stream (DVS)

Figure 1. The cloud accelerates the development value stream (DVS)

By aligning cloud capabilities with value-generating activities, enterprises maximize their ability to experiment quickly and deliver disruptive solutions to their markets.

There exist innumerable cloud strategies and implementation models, each of which has unique benefits. Among them are three common patterns that work together with SAFe to enable enterprise-scale agility. Here, they are characterized as Cloud for Infrastructure & Operations (I&O), Cloud for DevOps, and Cloud for AI (Figure 2).

Figure 2. Cloud usage patterns that enable business agility

Figure 2. Cloud usage patterns that enable business agility

Each of these patterns enables a unique DVS. Each DVS combines cloud capabilities with SAFe to deliver transformational innovation to a specific set of customers. Together, they scale operational efficiencies and business agility throughout the enterprise.

The first and most foundational of these patterns is Cloud for I&O.

Cloud for Infrastructure and Operations

Cloud for I&O is a *rehosting* pattern that often begins with migrating business solutions from on-premises data centers to the cloud. It automates the end-to-end infrastructure life cycle and delivers runtime environments that can be provisioned, scaled, and retired on demand. This significantly accelerates flow from infrastructure ÔrequestÕ to Ôretire,Õ alleviating common bottlenecks stemming from long service request lead times.

Figure 3 illustrates the Cloud for I&O pattern, depicting the key inputs, workloads, outputs, and architecture that increase the speed and agility with which core IT services and infrastructure are managed.

Figure 3. Cloud for I&O enables enterprises to rehost solutions for operational agility

Inputs ☰ The I&O cloud scenario begins with a service request for storage, network, or compute resources to support a software-based business solution’s deployment or production maintenance. Web-based, self-service catalogs expose secure, compliant resources directly to the requesting individual or team, allowing them to quickly fulfill their infrastructure needs.

Workloads ☰ The I&O cloud runs *production* workloads, such as the automated provisioning of application and database servers, executing application runtime containers, scaling hardware capacity for fault tolerance, administering security patches and maintenance updates, and retiring legacy infrastructure. Tasks that have historically been performed with long lead times by small teams of system administrators with elevated access privileges are performed in near real-time by

automated cloud services.

Outputs ☰ The I&O cloud delivers on-demand access to production resources and their configurations. This benefits [Agile Teams](#) by reducing, if not eliminating, service request lead times. It benefits business stakeholders by shifting the complexities of data center management to the cloud provider.

Architecture ☰ The I&O cloud consists of physical and virtual machines that provide highly elastic (horizontally and vertically scalable) storage, networking, and computing services. The cloud provider hosts this infrastructure and exposes it via self-service websites as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS) capabilities. The I&O cloud can be implemented in public, private, multi-cloud, and hybrid cloud configurations.

How SAFe helps ☰ Cloud for I&O requires that an organization's technical cloud strategy tightly aligns with its need to deliver continuous, sustainable business value. Datacenter migration, rehosting, and Ôlift and shiftÕ initiatives, therefore, must yield reliable hosting environments for business solutions and services that enable those environments to be configured, scaled, and reconfigured rapidly in response to changing business needs. SAFe enables this technology-business alignment in the following ways:

Customer centricity ☰ [Enterprise Architects](#), [Solution Architects](#), [System Architects](#), and [Product Management](#) collaborate closely to ensure that the I&O cloud is organized around value and designed to deliver the services that Agile teams and internal stakeholders need to realize significant productivity gains and economic benefits.

Strategic alignment ☰ Technical and business strategies remain in lockstep via [Strategic Themes](#), [Lean Budgets](#), [Enablers](#), and [PI Objectives](#) that are shared across the organization.

Incremental delivery ☰ Through adherence to [Principle #4](#) ☰ [Build incrementally with fast, integrated learning cycles](#), the I&O cloud delivers a continuous flow of value as it evolves, rather than delaying all functionality to the completion of a Ôbig bangÕ conversion.

Agile operations ☰ SAFe integrates I&O teams with [Agile Release Trains](#) (ARTs), fostering cross-functional collaboration and developing an organization-wide, Lean-Agile culture.

Cloud for DevOps

Cloud for DevOps involves shifting these I&O services ÔleftÕ into the delivery pipeline. This allows Agile teams to set up, use, and tear down secure, compliant cloud-based resources to support continuous solution innovation. With this pattern, solutions are *redesigned* using cloud-native technology and delivered as

modernized, next-generation products and services to be consumed directly by customers or integrated into larger solutions.

As depicted in Figure 4, this composite pattern incorporates Cloud for I&O as a critical enabler.

Figure 4. Cloud for DevOps enables enterprises to redesign solutions for continuous delivery

Inputs ☰ The arrival of one or more feature requests in the [ART Backlog](#) triggers this value flow, which initiates the [Continuous Delivery Pipeline](#). Typically, these requests are for features meant to optimize existing solutions for the cloud or deliver net-new experiences.

Workloads ☰ The DevOps cloud runs workloads that support the end-to-end solution delivery life cycle, such as Agile planning and design, low-code and no-code development, CI/CD, version control, application telemetry, security-as-code, and other activities described by SAFe's [DevOps Practice Domains](#). These workloads

automate the continuous delivery pipeline and provide Agile teams and ARTs the on-demand resources they need to build and sustain [Team and Technical Agility](#).

Outputs Ⓜ This pattern greatly accelerates feature delivery (often by orders of magnitude), generating high-quality, cloud-native solutions that offer competitive differentiation in the market. These are loosely coupled, containerized solutions that can be scaled and modified quickly to fuel continuous innovation. These outputs are used directly by end-users or integrated into other solutions, including [Large Solutions](#).

Architecture Ⓜ The I&O cloud makes storage, network, and compute environments available throughout the development value stream. This supplies Agile teams and ARTs with the infrastructure, applications, and [DevOps](#) capabilities required to fuel Continuous Exploration (CE), Continuous Integration (CI), Continuous Deployment (CD), and Release on Demand (RoD) activities.

How SAFe helps Ⓜ This pattern provides much of the tooling necessary to implement a continuous delivery pipeline and enable DevOps. However, effective DevOps involves more than automation. Combined with Cloud for DevOps, SAFe can ensure that organizations are working collaboratively across functions in the following ways to deliver value continuously:

A [CALMR](#) approach to DevOps Ⓜ Guided by systems thinking, SAFe provides an approach to DevOps that builds a culture of continuous delivery. Culture, automation, lean flow, measurement, and recovery are all elements of the DevOps mindset that SAFe keeps balanced throughout development, operations, and the business.

DevOps Practice Domains Ⓜ SAFe provides a comprehensive reference model that defines the practices and tools that fuel continuous delivery. These services should be implemented in the DevOps cloud and frequently assessed for quality using the [DevOps Health Radar](#).

[Agile Product Delivery](#) (APD) Ⓜ As one of SAFe's seven core competencies of the Lean enterprise, APD combines DevOps with customer-centricity, design thinking, and cross-functional Agile cadence. APD ensures that the DevOps cloud is calibrated to deliver valuable solutions in the shortest sustainable lead time.

[Built-In Quality](#) (BiQ) Ⓜ Much of the value of the DevOps cloud is in its ability to shift operations capabilities left to ensure high quality and fast flow through the delivery pipeline. SAFe's BiQ practices ensure that the DevOps cloud exposes services that enforce design, code, system, and release quality.

Cloud for AI

Cloud for AI enables enterprises to *reinvent* their solutions to achieve new levels of

competitive advantage. It provides the massive computing power required to fuel the development of sophisticated machine learning models, which drive intelligent, AI-based solutions that generate new value for the business.

Like Cloud for DevOps, Cloud for AI is a composite pattern that extends the capabilities of the I&O cloud (Figure 5).

Figure 5. Cloud for AI enables enterprises to reinvent solutions for the digital age

Figure 5. Cloud for AI enables enterprises to reinvent solutions for the digital age

Inputs □ This development value stream begins with the need for an AI-driven solution. It could be captured as an Epic, Feature, Capability, or Story and represents a business-focused opportunity to deliver an intelligent customer-facing solution, improve value stream efficiency, or gain insights about customers and markets.

Workloads □ The AI cloud runs workloads that enable the development of machine learning models through a highly iterative machine learning pipeline. This includes workloads that assist with acquiring and mapping data sets, developing and training machine learning algorithms, evaluating algorithms for accuracy, and

integrating and delivering the finished models. These workloads support the development of the four fundamental types of AI—supervised, unsupervised, reinforcement, and deep learning—through an iterative, collaborative machine learning operations (MLOps) process that involves data scientists and AI engineers.

Outputs Ⓛ This pattern produces intelligent, AI-driven solutions that address various needs inside and outside the enterprise. External customers benefit from more intelligent and engaging AI solutions, internal stakeholders benefit from AI-generated market insights, and Agile teams benefit from AI-driven enhancements to development and [Operational Value Streams](#).

Architecture Ⓛ The I&O cloud supplies the storage, network, and compute resources needed to execute the machine learning pipeline. These typically consist of big data environments and compute resources powered by graphical processing units (GPUs) that can handle the high data volumes and processing power required to develop ML models. Data scientists and ML engineers orchestrate these resources into an MLOps process that rapidly and iteratively advances the solution through the pipeline.

How SAFe helps Ⓛ As explained in the [AI](#) article, artificial intelligence offers enterprises the opportunity to increase competitive advantage and achieve better business outcomes significantly. But operationalizing and scaling this advanced, disruptive technology requires sophisticated cloud capabilities. SAFe helps organizations build and operate a reliable, value-aligned AI cloud in the following ways:

AI decision-making framework Ⓛ SAFe enables enterprises to make informed decisions about the AI cloud services that would serve them best through strategic alignment, customer centricity, continuous exploration, and empirical milestones.

Scaling AI capabilities Ⓛ Through [Principle #10 Ⓛ Organize around value](#), SAFe helps enterprises align their AI clouds to business needs. SAFe also facilitates the rapid dissemination of AI cloud expertise throughout the organization via cross-functional Agile teams, communities of practice, and numerous synchronization events that accelerate knowledge acquisition and sharing.

Enabling business agility Ⓛ SAFe orients the entire organization to a unified mission of building [Business Agility](#)—the ability to sense and respond to market opportunities with winning business solutions. Everyone involved in operationalizing the AI cloud, from data scientists and AI engineers to cloud architects and product managers, must be united in their focus on increasing overall business agility.

Managing big data Ⓛ Data is the lifeblood of the AI cloud. Still, the complex infrastructure required to manage that data—storage, distribution, and quality can create bottlenecks in the machine-learning pipeline. As described in [Big Data](#), SAFe provides guidance and tools for maintaining flow via Agile data teams, an iterative DataOps pipeline, portfolio-level support and funding, and federated data

governance.

Summary

Enterprises are increasingly turning to the cloud to create the agility they need to survive and thrive in the digital age. However, migrating enterprise systems from on-premises data centers to the cloud is not enough. They must also leverage the cloud to automate and accelerate value-generating activities across the organization.

Cloud for I&O, Cloud for DevOps, and Cloud for AI align an enterprise's cloud implementations with the development value streams that deliver disruptive digital innovation, combining powerful cloud services with SAFe principles and practices to enable true business agility.

Learn More

[1] Abdula, Moe, Ingo Averdunk, Roland Barcia, Kyle Brown, and Ndu Emuchay. *The Cloud Adoption Playbook: Proven Strategies for Transforming Your Organization with the Cloud*. Wiley, 2018.

[2] Orban, Stephen. *Ahead in the Cloud: Best Practices for Navigating the Future of Enterprise IT*. CreateSpace Independent Publishing Platform, 2018.

[3] Gartner. *Predicts 2022: The Cloud Moves from Technology Disruption to Business Disruption*, 2021. <https://www.gartner.com/doc/4008904>.

Last Update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT &
TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE,
INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS
HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Perfection is not attainable, but if we chase perfection, we can achieve excellence.

~Vince Lombardi

Value Stream Management

Value Stream Management (VSM) is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle.

Today, delivering digitally enabled solutions in large enterprises is often complex and spans many functional boundaries. Consequently, this structure can result in a fragmented delivery process with handoffs, cross-team dependencies, breakdowns in communication, and substantial delays. The purpose of VSM is to bring order to this chaos so that value-producing work can flow smoothly and continuously across the organization. The key to succeeding with VSM in SAFe is applying Lean thinking principles to every value stream.

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of eight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Details

In SAFe, Lean thinking is the foundation of VSM. Lean is an extensive body of knowledge aimed at improving operational efficiency by eliminating causes of delay. Lean thinking is based on the following five principles. [1]

1. Precisely specify value by specific product
2. Identify the value stream for each product
3. Make value flow without interruptions
4. Let the customer pull value from the producer
5. Pursue perfection

These principles provide a shared mindset for everyone involved in solution delivery. The remainder of this article describes each of the principles, how they enable VSM in SAFe, and who in the organization is primarily responsible for applying them.

1. Precisely Specify Value by Specific Product

This first principle of Lean thinking underscores the importance of understanding customers' needs and quantifying the value inherent in the products and services delivered. That value comes in the form of enduring [Solutions](#)—not temporary projects or initiatives—and is ultimately determined by the customer.

SAFe [Principle #1](#) **Take an economic view** informs VSM to define an overarching financial strategy for the value stream, including a bi-directional value exchange for each solution delivered (Figure 1).

Figure 1. Solutions require bi-directional value exchange

Figure 1. Solutions require bi-directional value exchange

Digitally enabled solutions are delivered through a SAFe [Portfolio](#) via one or more [Development Value Streams \(DVS\)](#) to external or internal customers. Those solutions support the delivery of market-facing products or services via [Operational Value Streams \(OVS\)](#). Each customer in the end-to-end flow of value derives tangible

benefits from the solutions they receive, such as increased productivity, customer satisfaction, or delight. Likewise, value is returned to the enterprise through cost savings or revenue. Although VSM can be applied to all value streams, SAFe uses VSM primarily to optimize the DVS.

The process of specifying value is aided by [Customer Centricity](#) and [Design Thinking](#). Being acutely tuned to the wants and needs of customers—and having a process to obtain continuous market feedback on solution ideas—ensures products and services deliver mutual, ongoing value.

Determining precisely how much value solutions deliver requires objective measurement. In SAFe, [Key Performance Indicators \(KPIs\)](#) are used to quantify the value solutions provide customers, and the economic benefit returned to the enterprise.

2. Identify the Value Stream for Each Product

Once the value of their products and services has been precisely defined, organizations must identify how they will develop and deliver that value to customers.

A DVS, or simply value stream in the context of this article, is the set of steps an organization executes to deliver a solution to the customer. The sequence of steps is unique to each value stream but generally involves incrementally defining, building, validating, and releasing functionality, as shown in Figure 2.

Figure 2. A basic development value stream in SAFe

Figure 2. A basic development value stream in SAFe

Development value streams are organized into SAFe portfolios, as Figure 3 illustrates. Each value stream delivers one or more solutions—*in the form of products and services*—to the customer. Budgets are allocated directly to value streams, funding the people, systems, and materials required to deliver those solutions fully.

Figure 3. A SAFe portfolio consists of one or more development value streams

Figure 3. A SAFe portfolio consists of one or more development value streams

SAFe Principle #2 [Apply systems thinking](#) guides the organization in identifying an entire value stream from request to release and regarding it as a single, integrated delivery system. This aligns efforts across the organization toward managing one shared value stream instead of many independently operated fragments.

Once the value stream has been identified, all people involved in its steps are organized to ensure smooth execution. Principle #10, [Organize around value](#) requires people with different skills to come together as a unified, enduring, cross-functional [Agile Release Train \(ART\)](#) to build the solution, as illustrated in Figure 4.

Figure 4. An ART contains all the people who work in the value stream

Figure 4. An ART contains all the people who work in the value stream

Large solutions—such as aircraft, automobiles, satellites, and medical imaging devices—typically require multiple ARTs and external suppliers to realize the value stream effectively. [Solution Trains](#) are established to organize the hundreds, sometimes thousands, of individuals needed to manage these complex value streams.

3. Make Value Flow without Interruptions

Once value streams are identified and teams are organized around the solutions they support, the value stream must be cleared of wasteful activities that delay solution delivery. Locating the sources of these delays is the first step toward improving value flow.

Value stream mapping accomplishes this by modeling the end-to-end sequence of activities and measuring performance at and between each step. As shown in

Figure 5, value stream maps help visualize the value stream and pinpoint major delay occurrences.

Figure 5. Value stream mapping identifies sources of delay

Figure 5. Value stream mapping identifies sources of delay

The next step toward accelerating flow is identifying the root causes of these delays and applying Lean-Agile methods to correct them.

SAFe identifies eight flow accelerators that help reduce interruptions and enable continuous value flow. They are listed below and are defined in detail in [Principle #6](#)
D Make value flow without interruptions:

- Visualize and limit WIP
- Address bottlenecks
- Minimize handoffs and dependencies
- Get faster feedback
- Work in smaller batches
- Reduce queue length
- Optimize time Ôin the zoneÕ

Remediate legacy policies and practices

These flow accelerators are time-tested approaches to addressing flow issues. How they are best applied depends on the nature of the issue and where it occurs in the value stream. The [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#) articles provide specific guidance for applying these accelerators at those levels to enable uninterrupted flow across the value stream.

4. Let the Customer Pull Value from the Producer

Value streams must deliver the right value to customers at the right time. This fourth principle of Lean thinking guides organizations to provide solutions that customers pull into the market based on their actual needs rather than pushing solutions based on what they think customers need (Figure 6).

Figure 6. Pull-based vs. push-based value flow

Figure 6. Pull-based vs. push-based value flow

Traditional, push-based delivery systems produce large, infrequent releases that typically deliver too little value too late. The solution is to create a pull-based flow, where small batches of work are prioritized and delivered quickly based on customer feedback and KPIs.

A simple yet powerful way to implement pull-based delivery is with a Kanban system. These are visual representations of the value stream (or segments of the value stream) that help regulate efficient flow through the system, as shown in Figure 7.

Figure 7. Kanban systems foster pull-based processing

Figure 7. Kanban systems foster pull-based processing

Kanban systems are inherently visual, pull-based and create smaller batches by applying WIP limits. SAFe encourages using [Team](#), [ART](#), [Solution Train](#), and [Portfolio](#) Kanban systems.

Further, solutions should be architected to support pull-based flow and on-

demand releases. Design these solutions to be loosely coupled, enabling individual components and sub-systems to be changed, tested, and deployed independently. This strategy avoids the high transaction costs and delays of Ôbig bangÕ change efforts. Accomplish this by architecting for [DevOps](#) and continuous delivery, using domain-driven design (DDD), containerization, and cloud-first methods. Also, feature toggles and blue/green infrastructure patterns effectively decouple deployment from release and allow releasing functionality on demand.

5. Pursue Perfection

VSM is an ongoing practice that continually optimizes the value stream in pursuit of maximum flow and quality. Although a perfectly efficient value stream is not achievable, the constant drive for perfection is a trademark of Lean thinking. It establishes a culture of continuous value stream innovation throughout the organization.

An organization pursuing delivery perfection must regularly assess its value stream against defined performance targets. Proper metrics, both quantitative and qualitative, provide a foundation for effective decision-making, replacing opinions with facts. [Flow metrics](#) (Figure 8) offer a comprehensive view of the work moving through the value stream, along with how quickly, predictably, and efficiently that work is flowing. These objective measurements help to identify the precise locations of delivery bottlenecks.

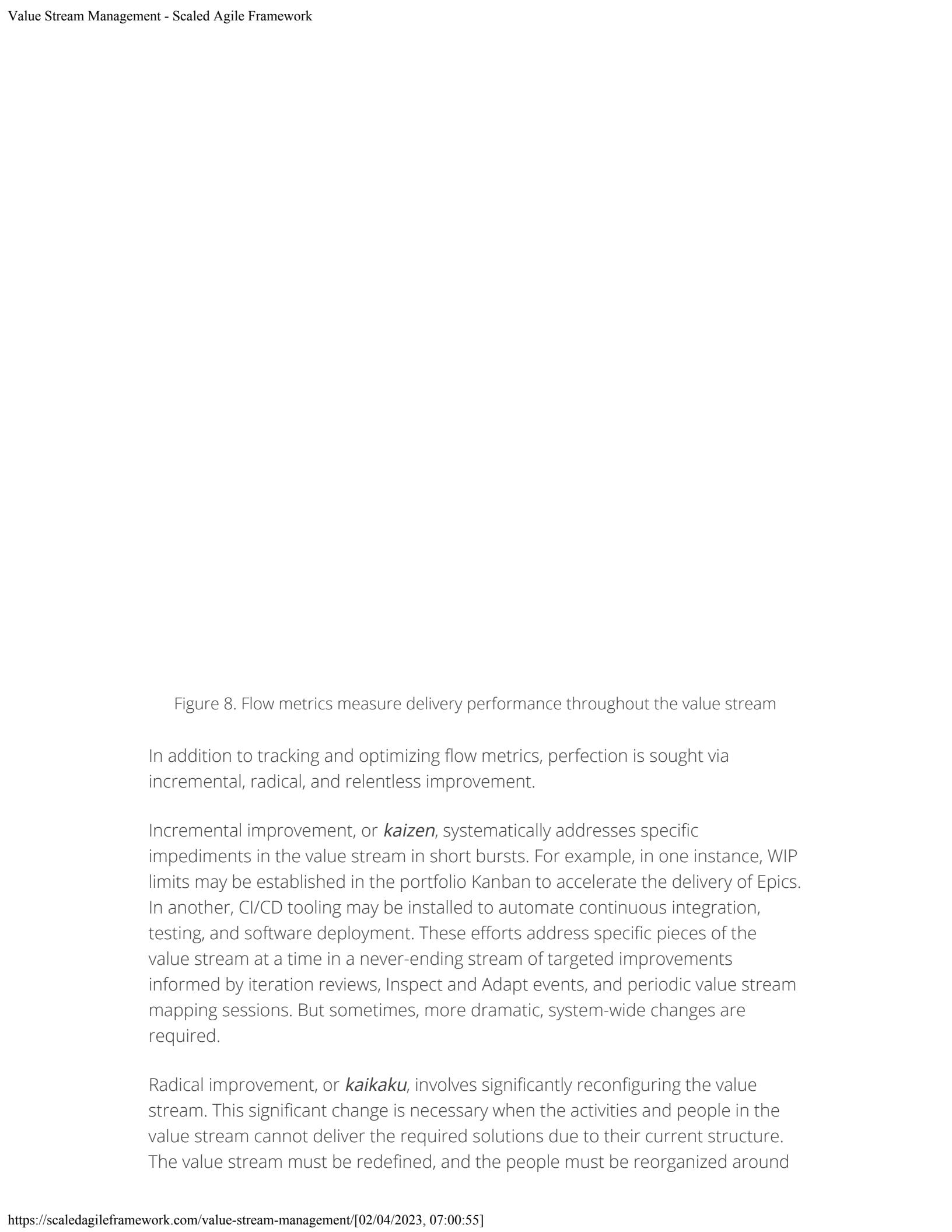


Figure 8. Flow metrics measure delivery performance throughout the value stream

In addition to tracking and optimizing flow metrics, perfection is sought via incremental, radical, and relentless improvement.

Incremental improvement, or *kaizen*, systematically addresses specific impediments in the value stream in short bursts. For example, in one instance, WIP limits may be established in the portfolio Kanban to accelerate the delivery of Epics. In another, CI/CD tooling may be installed to automate continuous integration, testing, and software deployment. These efforts address specific pieces of the value stream at a time in a never-ending stream of targeted improvements informed by iteration reviews, Inspect and Adapt events, and periodic value stream mapping sessions. But sometimes, more dramatic, system-wide changes are required.

Radical improvement, or *kaikaku*, involves significantly reconfiguring the value stream. This significant change is necessary when the activities and people in the value stream cannot deliver the required solutions due to their current structure. The value stream must be redefined, and the people must be reorganized around

the new flow of value. In these circumstances, organizations go through the process of identifying value streams and ARTs, replacing existing activities, and reconfiguring ARTs as necessary.

Organizations must continually identify inefficiencies in the value stream and make incremental and radical improvements when necessary to correct them. This persistent quest for perfection and bias for action signifies relentless improvement — a fundamental component of the Lean-Agile mindset.

Value Stream Leadership in SAFe

VSM is not easy. It entails the disciplined, ongoing application of the five principles of Lean thinking throughout each value stream. This, of course, requires the time, expertise, and dedication of key individuals.

Many organizations have appointed a Value Stream Manager — a role inherited from manufacturing — to serve in this capacity. A Value Stream Manager has the following characteristics:

Lean mindset □ They understand the principles of Lean thinking and how to apply them to improve value stream flow.

Business knowledge □ They understand the customer needs, market forces, and compliance factors that shape product strategy and define value metrics that guide product delivery.

Technical knowledge □ They understand which products, services, and supporting tools will produce the solutions with the most business value.

Process knowledge □ They understand the sequence of activities across the organization that turns ideas into valuable solutions.

Strategic influence □ They evangelize, support, enable, and secure funding for value stream improvements.

Tactical influence □ They identify needed improvements, mobilize teams, lead change, and regularly measure results.

A single Value Stream Manager may suffice for small, independent value streams; however, at scale, it is unlikely that any individual possesses all the necessary knowledge and skills. Therefore, in SAFe, a few key roles typically have joint responsibility for managing the value stream. As illustrated in Figure 9, ART and Solution Train triads carry out the tactical, output-oriented aspects of VSM, while **Business Owners** provide the strategic, outcome-oriented aspects. Together, they act as a team to provide the multi-faceted leadership required to manage the execution and evolution of value streams at scale.

Figure 9. Shared value stream leadership in SAFe

Figure 9. Shared value stream leadership in SAFe

In cases where a value stream manager *has* been designated, the person in this role would likely also serve as a Business Owner for one or more ARTs and assume the strategic VSM responsibilities listed above. If not, at a minimum, they would share the responsibility of managing the value stream with the individuals in these defined SAFe roles.

Triad Responsibilities

Triads align key individuals who possess business, technical, and SAFe expertise to a shared purpose of managing efficient delivery execution. [Product Management](#), [System Architects](#), and the [Release Train Engineer](#) (RTE) jointly guide ART execution, while [Solution Management](#), [Solution Architects](#), and the [Solution Train Engineer](#) (STE) jointly guide Solution Train execution.

Each role carries specific responsibilities within its domain of expertise—*as explained in its Framework article—but together, they have the added collective*

duty of managing the tactical execution of the value stream. Each triad applies the principles of Lean thinking to optimize its value stream for the most desirable outputs, as measured by SAFe's flow metrics.

Members of the triad work together, combining their strengths and spheres of influence, to define and architect solutions, technology stacks, and processes for continuous value flow, facilitate the identification of bottlenecks, and lead the organization through changes that accelerate value delivery.

Business Owner Responsibilities

Business owners are ultimately accountable for value stream outcomes. They provide the vision, strategy, governance, and leadership alignment required to operate and change value streams in alignment with evolving business needs.

Business owners apply the five principles of Lean thinking to optimize their value streams for the most desirable business outcomes. They define the KPIs that steer solution design toward realizing tangible economic benefits. Business owners then guide and support ARTs and Solution Trains through delivery in alignment with these KPIs.

To support relentless improvement, Business Owners sponsor value stream improvement opportunities identified by ART and Solution Train triads. When bottlenecks are so severe that they require radical changes to the value stream, Business Owners provide the strategic influence needed to define, prioritize, and implement systemic—sometimes disruptive—improvements that yield the greatest value to customers.

Summary

VSM is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle. SAFe enables VSM through its alignment with the five principles of Lean thinking and practices that optimize delivery efficiency across the organization.

VSM requires active, ongoing effort by leaders in the organization with business, technical, and process expertise, as well as strategic and tactical influence. In SAFe, these responsibilities are shared among the ART triad, Solution Train triad, Business Owners, and Value Stream Managers (if present) to ensure effective value stream management at scale.

Learn More

[1] Womack, James P., and Daniel T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York, NY: Free Press, 2003.

Last update: 27 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

- [GET SOCIAL](#)
- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



*Realize that
everything connects
to everything else.*

—Leonardo da Vinci

Business & Technology

Business and Technology describes the patterns that may be applied to realize Business Agility by applying SAFe principles and practices across the enterprise.

A successful Agile and SAFe adoption affects not just development but every part of the organization, including marketing, operations, support, finance, legal, HR,

Adjust Cookie Settings

security, compliance, and enterprise leadership. It's also clear that all business and technology domains can benefit by adopting a SAFe Lean-Agile mindset, Core Values, and principles and practices. Each area can use these foundations to discover how this new Agile way of working (see [Organizational Agility](#) for more detail) can improve outcomes in every business area.

Details

A Lean-Agile transformation will affect every part of the enterprise. As one area accelerates, it reveals bottlenecks in the others. In addition, every area of the business can benefit from agility. Thus, achieving full Business Agility requires applying the Lean-Agile Mindset, Values, and Principles to all aspects of the business:

A Lean-Agile Mindset The [Lean-Agile Mindset](#) forms the cornerstone of a new way of working and an enhanced company culture that enables Business Agility. It provides leaders and change agents with the tools needed to drive a successful SAFe transformation, helping individuals and enterprises achieve their goals.

Living the SAFe Core Values The four [Core Values](#) of *alignment, transparency, respect for people, and relentless improvement* represent the foundational beliefs that are key to SAFe's effectiveness. These tenets help guide the behaviors and actions of everyone participating in a SAFe portfolio. Those in positions of responsibility and authority can help the rest of the organization embrace these ideals by exemplifying them in their words and actions.

Applying SAFe Lean-Agile Principles SAFe is based on ten fundamental concepts that have evolved from Agile principles and methods, Lean product development, systems thinking, and observation of successful enterprises. Each is described in detail in an article by that principle's name. In addition, the embodiment of the principles appears throughout the Framework.

Agile Crosses the Chasm from Technology to Business



Our deep customer relationships, high-quality patient interactions, resilient supply chain, and agile operating model, all contributed to the strong quarterly performance.

—Karen Lynch, CEO CVS Health [CVS Q2 2022 Earnings Result Transcript \[1\]](#)

Twenty-five years ago, development bottlenecks started putting enormous business pressure on enterprises that depended on software. Typically, solutions were delivered late and over budget, often with unacceptable levels of quality. Traditional, stage-gated, waterfall development methods were unfit for the challenge. So, software practitioners began applying new, more Agile practices like Scrum and eXtreme Programming, as well as approaches based on Lean principles. This inspired the Agile Manifesto, ushering in a new age of increased team empowerment, experimentation, and faster feedback.

Over the past two decades, Agile software development Œcrossed the chasm [2],Õ as those early adopters invented the new practices and created demand for a broad ecosystem of products and services that support modern Agile software development. At the same time, software development remains on the critical path for most enterprises. Thus, the new Agile way of working has become the predominant method for enterprises facing the challenges of an increasingly digital age. But a new chasm and opportunity have emerged. Other technology and business domains now find themselves in the same position the software industry was in 20 years ago. Around the world, C-suite and shareholder discussions have identified agility as a strategic advantage to sustaining competitiveness in the digital age.

In short, Agile is here to stay for *technology and business*.

Patterns of Agile Business and Technology

SAFe transformations around the world are fostering emerging patterns that enable entire businesses to better learn, adapt and react to change. Figure 1 summarizes five patterns and trends for business and technology domains gathered from the experiences of practitioners leading the way.

Figure 1. SAFe Business and Technology Patterns

Figure 1. SAFe Business and Technology Patterns

Business-Enabled ART



“If you look right now in a team, you couldn’t distinguish what is a person from R&D, what is an IT guy, or from sales or marketing. They work together. This is really important to have this collaboration and to have the customer in the middle.”
CIO of *Porsche AG* and CEO of *Porsche Digital* [3]

A **Business-Enabled ART** is an [Agile Release Train](#) that includes the technical and business people needed to ensure the solution is aware of the business in which it operates and that it addresses the relevant concerns for the technology, business, and customer.

Agile Release Trains are purpose-built to develop and maintain the tech-enabled solutions the enterprise needs to thrive in the digital age. Naturally, most people on the ART are technologists who design, develop and integrate the software, systems, and physical components needed for state-of-the-art digitally-enabled solutions. Generally, ARTs take on one of three forms: stream-aligned, complicated subsystem, or platform ARTs, each with a specific set of responsibilities (see [Organizing Agile Teams and ARTs: Team Topologies at Scale](#).)

But it is the stream-aligned ARTs that have the potential to be more fully enabled to deliver business outcomes. Embedding the critical business functions from research, marketing, business processes enablement, contracts, technical support, legal, audit, and more enables the ART to provide a Ôwhole product solutionÕ to the customer (Figure 2).

Figure 2. Business Enabled ART

Figure 2. Business-Enabled ART

Many businesses have shown the value of this approach. Enterprises utilizing business-enabled ARTs achieve faster time to market, improved employee engagement, higher productivity, and better business outcomes.

The following considerations apply to establishing a Business-Enabled ART:

Ensure Critical Business Functions are contained within the ART ⚡ Embed the right people and skills needed to build the whole business solution in the ART. As an aid, the Value Stream Identification Workshop is critical to understanding what business functions are needed. A worked example can be found within the [Organize Around Value](#) step of the Implementation Roadmap.

Amplify Customer Centricity and product thinking ⚡ Business-Enabled ARTs deliver value directly to the customer. That requires technical and business competence. In addition, the core competency of [Agile Product Delivery](#) and the disciplines defined in [Customer-Centricity](#) and [Design Thinking](#) are even more critical.

Focus on the intended business outcomes ⚡ Identifying the desired results for the user and the business helps provide the objectives and focus for the ART. This

pattern is best built with the people who already have frequent contact with the customer or, in some cases, are the customer. Utilizing this shared knowledge amplifies the effectiveness of OKRs, KPIs, and PI Objectives.

Example: A Data-Informed Decisioning Business-Enabled ART

An existing ART is developing a set of data-informed digital care plans for patients with critical illnesses. After a year of applying SAFe, they identified a consistent set of dependencies with data analysts and a group of internal physicians. They work with the LACE team to re-run the Value Stream Identification Workshop alongside the leaders of those business areas. As a result, the ART added the people they interact with the most. The data analysts join the existing Agile teams, while an existing physician joins the ART Product Management team. A few PIs later, they realized accelerated flow in every flow metric and their OKRs, including an increase in employee NPS. The Business-Enabled ART demos their improvements at all hands, resulting in more ARTs in the organization moving to the Business-Enabled pattern.

Agile Business Train

An Agile Business Train (Figure 3) is a further step towards Business Agility. An Agile Business Train contains one or more SAFe [Operational Value Streams](#) and all the ARTs needed to define, build, deploy, operate, and commercialize a complete business solution. An Agile Business Train could include multiple ARTs and Solution Trains, all aligned to a shared mission, vision, and cadence. Participants plan, commit, ideate, deliver, and continually optimize the business solution together.

Figure 3.

Figure 3. The Agile Business Train

Building these partnerships across hierarchical functions with cross-functional Agile teams enables the right business processes and business solutions to flex, grow and flow with the combined expertise of the workers involved.

Here are a few tips for designing an Agile Business Train:

Understand both Value Stream flows ▷ Mapping how new solutions get developed and the operational value flow, including all the people involved, is critical for an Agile Business Train. Organize Agile teams around the ability to operate, commercialize, and support existing users, as well as how to identify and build new solutions.

Application of the Core Competencies ▷ The core competencies of Team and Technical Agility, Agile Product Delivery, Continuous Learning Culture, Organizational Agility, and Lean-Agile Leadership all apply. While technical practices differ by function (a nurse's technical practice is different from a hardware engineer's practice is different from a retail store worker's practice), all practices require focus and improvement through a relentless improvement mindset.

Cadence-based planning and PDCA ⚫ Aligning on a cadence with operation and development teams enables operational improvements to scale alongside systematic technology improvements.

Example: Home Health Business Train

An Agile Business Train comprises case managers, home nurses, data scientists, developers, regulatory specialists, physicians, medical device technicians, and researchers within a mid-sized healthcare organization. The operational value stream they combined to deliver was home hemodialysis care for a particular geographical region of patients across multiple care facilities and patient homes. Each team was cross-functional, enabling every team to utilize SAFe Team Kanban methods and PDCA learning cycles to provide the best patient care. If a home medical device indicated errors, device technicians would immediately respond and help the patient or the nurse work through the issue. The blended train utilized PI cadences and demos to continually update their procedures and care plans. They learned together from the data gathered across the teams, keeping more patients healthier and increasing the satisfaction of the patients and practitioners.

Figure 3.

Figure 4. An example Business Train dedicated to Home Health Care

Combined Portfolio

In some instances, due to significant coupling between Development and Operational Value Streams, it is helpful to manage those value streams in one portfolio. A **Combined Portfolio** is a type of SAFe portfolio that includes both Development and Operational Value Streams. It leverages knowledge of the operational costs, the performance of the operational and development value streams, revenue, and market research to support strategic decision-making. The Combined Portfolio takes the concept of Systems Thinking beyond just value streams and enables a more global optimization.

A Combined Portfolio enables an organization's executives to communicate and execute the complete strategy. It is then easier to align new development

opportunities with how operational work is produced, through funding, strategy, and the flow of epics through the Portfolio Backlog (Figure 5).

Figure 5. Elements of a Combined Portfolio

Figure 5. Elements of a Combined Portfolio

The following considerations apply when establishing a Combined Portfolio:

Funding ☐ The funding for the two types of value streams gains value in budgeting discussions and horizon allocations. In either case, funding must be aligned on both ÒsidesÓ of the portfolio to ensure the ability to deliver value together. Digital technologies may provide opportunities to move workers into a digital role supporting the same end customers or require combined funding of value streams that enable innovations on the operational side to scale the business fully.

Strategic Themes and Portfolio Vision must reflect both ☐ A combined portfolio may naturally have a significant portion of strategic work cut across both development and operational value streams. Strategic themes keep the portfolio aligned with the needs of the enterprise and maintain connections across portfolios. The combined portfolio utilizes OKRs to align the outcomes across development and operational value streams with the themes.

Work from a common Portfolio Backlog ☐ Combined portfolios utilize a single Portfolio Kanban system to help steer the flow of their high-profile work, supported by stakeholders from operational and development value streams. Prioritization is done in concert, optimizing for funding solutions in concert with the operational value stream(s).

Example: Retail Business Combined Portfolio

An organization utilizing digital experiences to drive users towards a convenient, brick-and-mortar retail store presence applied the Business -Enabled ART and Business Train patterns within an Operational Value Stream. The resultant Combined Portfolio includes Platform ARTs, Business Trains comprised of store employees and regional managers. In addition, it includes Business-Enabled ARTs comprised of cross-functional teams of marketing specialists, data scientists, regional compliance specialists, communications strategists, and behavioral scientists. Portfolio Syncs and participatory budgeting events consider leading indicators on the operational and development value stream side. For instance, a new Omnichannel communication platform MVP is tested within a region of retail stores. Retail associates provide feedback directly to the development value streams on new behaviors of in-store customers. The retail associates know about the latest release from their store managers, and the regional store managers bring customer reactions to the next Portfolio Sync. One discovery was that 15% of customers felt that the new location messaging was overly invasive. And yet 85% appreciated the immediate savings. Therefore the related epic continues, now emphasizing less intrusive and more welcoming messaging.

Agile Business Function

Agile Business Functions apply Lean-Agile methods and SAFe principles to streamline business operations and create transparency in value delivery. This pattern can be applied in all functional areas of the enterprise. This pattern helps improve the flow of value throughout the organization, increases workers' motivation and engagement, and provides transparency for the operational leadership. Agile teams in these functions may operate within the functional area or department or provide shared services to ART and other functions, as illustrated in Figure 6. And as with all things Lean and Agile, Lean-Agile Leaders provide the foundation needed to establish, operate and sustain the new way of working.

Figure 6. Example of an Agile Business Function

Figure 6. Example of an Agile Business Function

The following considerations apply when building effective Agile Business Functions:

Agile Teams **D** People within the function are organized into Agile Teams, which are cross-functional by design. Even though the department accomplishes one primary function, multiple areas of expertise contribute to the function. Agile Teams include those different skills and follow an appropriate team process, whether SAFe Scrum, SAFe Team Kanban, or a hybrid.

Interaction with the rest of the organization **D** As noted, business function Agile teams may be embedded in a cross-department Agile Release Train or operate entirely within the functional department. In the former case, the team may be more loosely coupled with its functional department and operate more under the auspices of the train's process. In the latter case, the department establishes mission-appropriate interaction protocols with other parties and manages service capacity allocations to prioritize and address demand.

Example: Agile Data Science and Analytics

Data science is often organized into a department separate from IT. Due to exponentially increasing demands for data, data analytics, and the scarcity and specialty skills involved, it may be impractical to embed these functions fully into the various ARTs. In one enterprise, leadership transformed the department into Agile Teams to increase support for these critical workers. They established a functional-wide backlog and kanban fed from ARTs, teams, and other organizational functions. They proactively allocated time for team members to spread their skills to other engineers and architects. The data scientists can also easily share their insights and be available for help, eliminating redundant hires in other areas of the organization. At times, an existing ART may require more fully embedded analysts (for example: a Behavior Change ART utilizing daily data feeds to create motivational messaging and gamification within an existing application). In that case, the department staffs a full-time Agile team dedicated to that ART while maintaining its departmental structure and the ability to gather insights from the larger pool of people and data.

Agile Executive Team

The C-suite teams dominating the Forbes lists prove that consistent shareholder and revenue growth is tied to the passionate and constant pursuit of a clear mission that aligns the organization and stakeholders in strategy and execution. The **Agile Executive Team** is a construct for aligning senior leadership in the form of Agile team that exhibits the mindset, values, principles, and practices of agility.

The Agile Executive Team construct creates structure and process that helps the organization's top leaders dedicate sufficient time to critical alignment, impediment removal, and effective management of strategic WIP implemented by the portfolios.

Figure 7. An Example Agile Executive Team

Figure 7. An Example Agile Executive Team

The Agile Executive Team resembles a SAFe Agile Team in many ways:

- Sized like a typical Agile team (10 or less)
- Has a ScrumMaster/Team Coach and a Product Owner
- Maintains a single backlog
- Frequently synchronizes within the team and with the rest of the organization.

Here are some aspects of the Agile Leadership Team that are critical to its successful operation:

Exemplar behaviors and practices ☰ This team exemplifies alignment, transparency, respect for people, and relentless improvement. They speak as a Lean-Agile leadership team even when speaking as individuals. They mitigate silo-based thinking and create an environment that encourages celebrating customer outcomes and learning. They encourage organizing their direct reports into Agile Teams. This practice allows C and VP levels to model practices and behaviors for the thousands of people reporting to them.

PO and Team Coach role ☰ Someone assumes the role of a PO and maintains the backlog. In this case, however, prioritization and decision-making is the collective responsibility of the entire team. An SM/TC facilitates a productive, collaborative process, allowing the team to express multiple opinions and converge on an agreed-

upon priority list and effective decision-making.

Planning and Synchronization ☰ The Agile Executive Team may have a daily sync or perhaps meet two to three times a week on an established schedule. The syncs include aligning each executive's interactions with their direct reports and others in their respective areas of responsibility. Every week or two, the team holds a more detailed planning session.

Periodic Strategic Alignment ☰ This team has the ultimate responsibility for strategy and execution, and they must allocate the time to evolve a strategy continuously. They may also meet periodically to address systemic impediments that require direct leadership support.

Interaction with Portfolios ☰ The team regularly communicates the enterprise's strategic intent and provides direction to solution portfolios. They actively receive feedback on the systemic impediments to enterprise flow and assist in removing them.

Example: Medical Device Organization

An Agile Executive Team is formed within a medical device organization focused on penetrating existing markets in more innovative, data-empowered ways. The CEO, VP of Regulatory, Chief Product Officer, VP of Engineering, and CFO meet together after their personal tipping point. They've recognized that the company's employees are acting in silos, and they must visibly show up as a team to meet their strategic goals of utilizing new Agile methods to increase delivery in their highly regulated environment. The CEO shares openly that they will need good facilitation to help keep their conversations on track. The VP of Engineering recommends utilizing the same model they expect of their teams. They pull in a trusted Scrum Master to maintain confidentiality when needed and keep them focused. Within three months, the organization's employees follow their example and break down old barriers. As a nice side effect, the Agile Executive Team is now meeting once a month with their direct reports, also forming as Agile Teams, to open communication and take action about blockers to the organizational flow. This has already resulted in a value-focused, cross-functional assessment of their entire set of regulatory policies and procedures, eliminating significant areas of waste and increasing employee engagement.

Building the Agile Enterprise

Agile Teams and Trains work together to deliver valuable business solutions regardless of whom they contain. The patterns above, alongside SAFe's principles

and values, unleash the potential of Enterprise-wide Business Agility. Utilizing these patterns, powerful outcomes are accelerated, which include:

- An accelerated flow of Business Solutions through the organizational systems.
- Removal of barriers to decision-making and action.
- Improved alignment, focus, and transparency across the enterprise. Everyone Ÿsees, knows, and acts together in the business and customer's interest.
- Alignment in execution, removing any historic technology **vs.** business mindset. All teams advance the business.
- Elimination of redundant roles and waste in the system.
- Increased empowerment, authority, and accountability throughout the business.

Welcome to the new age of the Agile business.

Learn More

[1] CVS Health. ŸQ2.2022 Earnings Results Transcript. [investors.cvshealth.com, https://investors.cvshealth.com/investors/events-and-presentations/default.aspx](https://investors.cvshealth.com/investors/events-and-presentations/default.aspx). Accessed 2 Nov. 2022

[2] Moore, Geoffrey. *Crossing the Chasm*. 1991.

[3] Scaled Agile, Inc. ŸDesigning the Digital Future with PorscheÓ <https://www.youtube.com/watch?v=BjM8vZNujCc>. Accessed 3 Nov. 2022

Last Updated: 15 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Artificial
Intelligence, deep
learning, machine
learning—whatever
you’re doing if you
don’t understand
it—learn it. Because
otherwise, you’re
going to be a
dinosaur!*

—Mark Cuban,
American billionaire
entrepreneur

Adjust Cookie Settings

Artificial Intelligence in SAFe

Artificial Intelligence (AI) is a term used to describe a wide range of smart machines capable of performing tasks that typically required human intelligence. AI can be applied at all levels of SAFe to build intelligent customer solutions, automate value stream activities, and improve customer insights.

It is a technology that can revolutionize solutions developed by SAFe organizations and has the potential to dramatically influence the operational and business models of enterprises as well.

The Growing Influence of AI

Every day, more and more aspects of our personal lives are aided and supported by intelligent systems. Applying for a loan, flying on an airplane, shopping in an online store, and scheduling a doctor's visit are just a few limitless examples of daily activities using Artificial Intelligence (AI). Further, consumers are not the only beneficiaries of AI-powered solutions. Enterprises use AI to understand their customers better and create better products and services. Banks use AI to detect fraud and money laundering. Governments provide services to their citizens using AI. Even military and national security systems have incorporated this emerging technology.

The trend is clear:

A startling 72% of executives believe that AI will be the *most* significant business advantage of the future [1].

According to a PwC survey performed in 2020 [2], 86% of respondents said AI would become a "mainstream technology" at their company in 2021.

By 2025, 70% of organizations will have operationalized AI architectures [3].

Critical factors have accelerated the rapid growth of these ever-more intelligent technologies. Significant advancements in general-purpose hardware, cloud technology, and optimized infrastructure, such as Graphics Processing Units (GPUs) [4], provide unprecedented computing power that supports a different magnitude of AI capabilities. Concurrently, new learning mechanisms and AI architectures have emerged that expand the frontier of business problems solvable by AI. In addition, the COVID-19 pandemic forced many enterprises to accelerate their multi-year digital transformation plans into weeks or months, thus bringing a robust big data foundation to a broad range of AI-powered business opportunities. As a result, according to Gartner's research, over 80% of Global

2000 companies have launched AI initiatives. Many of those solutions are in production and on the market today.

It is increasingly evident that every organization must seriously evaluate and leverage AI capabilities in their next generation of products and services to stay competitive.

This article provides a foundational context for applying AI in the enterprise and how SAFe can accelerate the successful adoption of this advanced technology. Key topics covered include:

AI as a Competitive Advantage highlights the nature of AI's business value and what is required to build a sustained advantage over the competition.

Applying AI to Achieve Better Business Results outlines typical application opportunities for AI within operational and development value streams and capabilities that enterprises can directly embed in the products and services delivered to the end user.

Understanding the Fundamental Types of AI describes common AI concepts and architectures that technology and business leaders must understand to apply AI within their enterprise.

Success Factors for AI Initiatives with SAFe describes essential practices and a decision framework for successful AI initiatives with SAFe.

AI as a Competitive Advantage

AI opens various possibilities to extend existing solutions and make them more valuable and scalable. More importantly, it opens a frontier for new solutions and qualitatively distinct capabilities that benefit the customer and the business in a new way.

But where exactly is AI most helpful?

Conventional software solutions are good at addressing well-understood problems; such solutions execute a limited number of scenarios based on a predefined list of rules. However, many organizational workflows and customer scenarios involve parameters that cannot be accounted for via conventional, preprogrammed means. AI addresses many of these most complicated scenarios and turns them into viable business opportunities. The landscape of potential AI applications within an enterprise is broad and diverse (Figure 1).

Figure 1. AI dramatically extends the range of business solutions

Figure 1. AI dramatically extends the range of business solutions

For example, a bank may utilize AI to automate customer service, identify suspicious account activity, understand customers' needs better and offer appropriate products, extract essential insights from customer feedback and social media, facilitate regulatory compliance, and so forth.

As specific AI applications become more commonplace, companies are moving beyond basic AI solutions to find new opportunities where complex problems exist that can be solved with the help of AI. The ultimate goal is to create innovative business value ahead of the competition [5]. But creating successful AI solutions is no easy task. While AI is still software, the nature of machine learning software development is fundamentally different. Many organizations don't know how to leverage an AI opportunity effectively, even when they devote significant resources to the problem. SAFe provides an operating model that helps identify opportunities, validate new capabilities, and translate them into valuable, working solutions that benefit the customer. The first step is understanding how AI helps improve an organization's outcomes.

Applying AI to Achieve Better Business Results

Generally, an organization's opportunities for leveraging AI lie in three areas (Figure 2):

Figure 2. The three primary ways to apply AI in an organization

Figure 2. The three primary ways to apply AI in an organization

1. **Increasingly Intelligent Customer Solutions.** More and more intelligence is embedded in the products and services we access every day. Common examples include self-driving vehicles and driver-assisted capabilities, facial recognition on our phones, personalized product recommendations, and smarter devices powering the Internet of Things (IoT). In addition, many AI-powered functions are now directly embedded in our mobile and desktop computer software applications.
2. **Improved Operational and Development Value Streams.** Organizational processes are ripe for AI applications to make an internal enterprise process more productive or introduce new capabilities. AI solutions can be applied to both

Operational and **Development Value Streams**. Inventory and demand management, personalized consumer experiences, and fraud detection are examples of how AI can power operational workflows. In the case of development value streams, companies can use AI to navigate the solution context, analyze production data, identify optimal parameters for user scenarios, and facilitate effective testing.

3. **Customer Insights.** AI can help organizations identify new business opportunities, learn more about the customer, and extract market insights that create entirely new offerings on an even broader scale. In this latter case, AI supports launching new initiatives via the BAVS (the **Business Agility Value Stream**) that were otherwise undiscoverable.

Understanding the Fundamental Types of AI

Artificial intelligence (AI) is a wide range of smart machines capable of performing tasks that typically require human intelligence. The potential applications represented by AI are extensive and affect almost every facet of business and consumer life. Many of today's AI systems are based on Machine Learning (ML). ML-based solutions are designed to improve based on experiences and data. However, some AI architectures do not involve Machine Learning and instead are based on a comprehensive set of static rules that encode some complex reasoning. Figure 3 provides a typology of various AI and machine learning approaches. It also illustrates some of the capabilities that these technologies enable. However, some illustrated capabilities in Figure 3 can be built with more than one AI approach, and a combination of methods may also be applied.

Figure 3. The landscape of Artificial Intelligence

Figure 3. The landscape of Artificial Intelligence

As shown in this graphic, there are three major categories of ML, differentiated by how the *learning* is achieved. All three approaches involve three primary components: the data, the learning algorithm, and the learning model, as Figure 4 illustrates.

Figure 4. Three critical components of machine learning

Figure 4. Three critical components of machine learning

Most generally, Machine Learning can be described as one of four types, as described below.

Supervised Learning

Supervised learning utilizes training data to *teach* the model how to produce the desired output (Figure 5). The training data must contain the inputs and the desired outputs as labels. The learning algorithm runs inputs through the model, compares them with the labels, and computes the model output.

Figure 5. Supervised learning

Figure 5. Supervised learning

The algorithm adjusts model parameters and repeats the process until it reaches a sufficiently low number of errors. It is called ***supervised*** because the desired outputs are supplied alongside the inputs and are used to ÔsuperviseÕ or ÔguideÕ the learning process. Unless the data initially includes both the inputs and the labels, it requires a ÔlabelingÕ process ***before*** training the model.

Supervised learning can help detect known patterns (fraudulent transactions, spam messages) and data categorization (image recognition, text sentiment analysis). In some instances, the output data may be readily available or easily attainable in an automated manner (such as a customer name alongside the profile photo for face recognition or a five-star rating score next to the product review text for sentiment detection; the situation often referred to as ***self-supervised learning***). Identifying such facets of data opens excellent opportunities for applying supervised learning to organizational processes.

Unsupervised Learning

Unlike the previous approach, Unsupervised Learning does not utilize any feedback mechanism. Instead, it extracts valuable information merely by analyzing the internal structure of the data.

Figure 6. Unsupervised learning

Figure 6. Unsupervised learning

Unsupervised learning has a significant advantage because the input data doesn't need to be labeled, allowing the learning algorithms to use vast volumes of data. This approach supports easier scaling of the unsupervised learning-enabled capabilities.

This type of AI algorithm is applied to data clustering, anomaly detection, association mining, and latent variable extraction tasks. These processes partition the data by similarity and establish existing relationships within data to be used by other solution capabilities or functions. Some common use cases of such tasks are customer or product segmentation, similarity detection, and recommendation systems. Unsupervised learning can also be leveraged as a link in a broader chain of a supervised learning process to extend data labeling to unlabeled datasets.

Reinforcement Learning

Reinforcement Learning is similar to supervised learning because it also involves a feedback mechanism that verifies the model. However, the feedback does not rely

on labeled data in this case. Instead, the system *acts* in a particular *environment* and is supplied with a *reward function* that helps the model learn what action leads to successful outcomes. Therefore, the learning algorithm generates exploratory activity and selects scenarios that lead to the highest reward.

Figure 7. Reinforcement learning

Figure 7. Reinforcement learning

Reinforcement learning finds applications in robotics, gaming, decision support systems, personalized recommendations, bidding and advertising, and other contexts where simulated exploratory behaviors can be evaluated in terms of their value.

Deep Learning

Deep learning is the label for machine learning models based on artificial neural networks (ANN). Deep learning can be effectively applied to supervised, unsupervised, and reinforcement learning and, in many practical tasks, has produced results comparable to or surpassing human expert performance. An artificial neural network is loosely modeled after the structure of neurons in the brain. An ANN has *inputs*, *outputs*, and consists of a connected set of *neurons*. An example of such a model could be a neural network that accepts pixel colors in an

image as an input and determines what type of object is in that image as an output.

Figure 8. A deep neural network applied to pattern recognition

Figure 8. A deep neural network applied to pattern recognition

Every connection has a specific *weight* that either strengthens or inhibits the signal. When all the connectors leading to a particular neuron convey a sufficiently strong cumulative signal, the neuron *activates* and transmits the signal to other neurons further downstream.

A neural network with multiple hidden layers is called a *deep neural network* and is the foundational architecture for deep learning.

Generative AI

Generative AI is a subfield of artificial intelligence that focuses on creating new content and experiences through machine learning algorithms. It has the potential to revolutionize the way businesses operate and create value. Both businesses and the general public became aware of generative AI virtually overnight with the

release of ChatGPT in late November 2022. Initially offered as a free tool, ChatGPT gained over a million users in its first five days and exceeded 100 million users by the end of January 2023.

Generative AI differs from other types of AI in its focus on creating new content and experiences. Unlike other AI applications, such as supervised learning and reinforcement learning, generative AI algorithms are trained to generate original output, such as images, videos, or text. This makes generative AI a powerful tool for businesses looking to automate creative tasks, generate digital assets, and drive innovation. While other AI applications are designed to recognize patterns in existing data, generative AI focuses on producing new, unique content not seen in the training data.

One major application of generative AI is to generate realistic product images or write compelling ad copy. This can significantly reduce the time and resources required for businesses to produce high-quality digital assets. Another area where generative AI can have a significant impact is in automating repetitive tasks and processes. For example, generative AI algorithms can be trained to perform tasks such as invoicing, data entry, and report generation, freeing up employees to focus on higher-value tasks. Generative AI also has the potential to drive innovation and new product development. By combining generative AI algorithms with other technologies, such as simulation and computer-aided design, businesses can quickly generate and test new product concepts, helping to speed up the development process.

However, it's important for businesses to be aware of the potential ethical implications of generative AI, such as the creation of fake, misleading, or biased content. It's critical for organizations to have strong governance and ethical frameworks in place to ensure that their use of generative AI aligns with their values and mission. Precautions must also be taken to avoid exposing customer data, personally identifiable information (PII), or sensitive company data to public generative AI tools.

(Note: The preceding description of generative AI was generated by ChatGPT! [6])

Success Factors for AI Initiatives in SAFe

Developing and delivering successful AI solutions is a task challenging to many organizations. The following factors are critical in establishing a productive solution development process for AI systems.

Apply a Clear AI Decision-Making Framework

Many AI initiatives fail to deliver the better results promised when advocates propose investments in this technology. This failure to meet expectations is often caused by poor decision-making about how and why AI will be used. Organizations often want to Ôdo AIÕ because Ôeveryone else does,Õ without understanding the effort required to operationalize and scale this technology, its impact on the organization, or even if it will produce the intended benefits.

SAFe organizations already have powerful tools to make better decisions on the appropriate use of AI. A few of these are highlighted in Figure 9.

Figure 9. AI decision-making framework enabled by SAFe

Figure 9. AI decision-making framework enabled by SAFe

1. Alignment with strategy ensures that AI initiatives pursue beneficial outcomes for the business. Aligning the AI roadmap with the [Portfolio](#) strategy is an essential step in this direction. Some AI initiatives may require more financial support; others may need to be repurposed or canceled based on a progressively assessed economic viability. Productively managing the spending requires [Lean Budgeting](#). Using the [Portfolio Kanban System](#) and a [Lean Business Case](#) helps

establish better alignment with strategy. Additionally, [PI Planning](#) provides the foundation for recurrent alignment of AI strategy with the actual implementation.

2. Customer centricity is critical to ensuring that an AI initiative is actually solving a customer problem. For example, intrinsic AI capabilities (such as image recognition or natural language processing) must be appropriately integrated into a favorable customer scenario. Explicitly defining the customer problem is an important step and hugely benefits from applying [Design Thinking](#) to AI capabilities.

3. Continuous exploration paves the pathway to a successful AI-powered solution. Solution development always contains a significant degree of uncertainty. In the case of AI, however, the level of uncertainty is exceptionally high in creating the right solution and implementing it correctly. This is where the [SAFe Lean Startup Cycle](#) is very useful. Creating a clear business hypothesis, building an AI MVP, and validating it against suitable measures are at the core of successful exploration.

4. Empirical milestones guide the development of a successful AI solution. AI capabilities must be continuously integrated with the rest of the solution throughout the incremental development process. Solution increments are used to elicit important customer feedback.

Organize to Deliver AI Solutions

Organizing around value is a critical enabler of flow and is one of the SAFe operating [principles](#). In the case of AI-powered solutions, it has specific implications for SAFe organizations. We've observed three stages of maturity and organization evolution, as Figure 10 illustrates.



Figure 10. Observed organizational models for AI maturity in SAFe

Stage 1. The first step is to build a critical mass of AI expertise. This step includes hiring and developing skills in AI and ML, and the accompanying skills in data science, data analytics, and data engineering. This centralized team engages with subject matter experts. It may take on building and deploying some initial proof of concept, pilot projects, or even an MVP for an anticipated AI initiative. This stage seeds the enterprise with AI thinking and shows the potential for extending the range and types of AI applications.

Stage 2. It quickly becomes apparent that these new capabilities need to be developed and integrated into the enterprise's product offerings. Integrating AI capabilities into existing products requires the engagement of the ARTs responsible for enterprise solutions. As a result, the focus of the central AI team moves from developing AI capabilities to enabling ARTs to do so. At this stage, the team may embed some subject matter experts on board each train to streamline the process. Typically, the centralized function also continues to develop certain types of AI capabilities, particularly those that cross value streams and cause the creation of more mature, enterprise-level data.

Stage 3. With time, more AI functions get embedded as a part of the Agile Release Trains. Including these skill sets on the ART helps manage integration risks, establishes a more robust delivery pipeline, and fosters critical knowledge. The decentralized effort, however, typically benefits from ongoing centralized expertise. A central AI architecture team can continue to explore new AI technologies, implement portfolio and enterprise cross-cutting initiatives, and provide organizational governance that ensures proper management of customer data and AI's ethical and business implications in the marketplace.

Across all three stages, the AI group must involve engineering and business professionals. This is vital for the organization to identify and leverage productive opportunities for AI-enabled solutions.

Proactively Manage Data

Data is a critical link in most AI implementations and, when handled poorly, leads to a failed AI initiative. Every organization must establish an effective process around data management.

Data is critical to AI in a couple of ways. First, most AI systems require a *training dataset* that must be large enough and thus referred to as Big Data for the AI model to learn to perform its task correctly. For example, a fraud detection system may require tens or even hundreds of thousands of transactions until it learns to recognize the fraudulent ones amongst them reliably. To validate whether the system was successfully trained, a whole separate dataset is needed – the *testing dataset* that must be readily available.

But the great challenge comes not only from the size of big data. An even more significant challenge is that data seldom exists in a ready-to-go, consumable form. Data is often dispersed across multiple, disjointed sources within and outside the enterprise. Additionally, the actual format of the data may be unconsumable. It may lack some critical attributes, without which training is impossible.

Any AI approach will depend on a portfolio strategy that recognizes that a systematic approach to acquiring and normalizing big data is virtually always the precedent for an AI application. This includes validating assumptions about data as a first order of the business in every AI initiative. Figure 11 illustrates some central elements of how SAFe guides enterprises on this parallel journey.

Figure 11. The Big Data cycle

Figure 11. The Big Data cycle

Build Organizational Competency Around AI

Artificial Intelligence represents a novel type of technology to many organizations. And even those companies with some AI initiatives in flight don't have the right expertise applied in the right place. AI is a complex technology that requires a good understanding of AI capabilities and potential business applications' landscape. Navigating critical opportunities requires business and technology people in an organization to work collaboratively to develop AI competency.

SAFe emphasizes the importance of the constant interaction of business and technology professionals. This interaction is achieved through multiple means, including strategic effort around the portfolio kanban system, events like PI planning, [System](#) and [Solution Demos](#), and [Inspect & Adapt](#). Additionally, various continuous exploration activities, design thinking, and work definitions leverage the overlap between business and technology representatives. In the case of AI, all these techniques need to be utilized even more rigorously, as disconnects may be very costly.

Establish an AI Solution Path

Driven by the flow of [Business Agility](#), an AI solution advances through the following logical steps:

Figure 12. Example AI solution path

Figure 12. Example AI solution path

Define Δ This step involves identifying the business opportunity and understanding the customer’s need. The initial assessment determines how suitable AI architectures could support the business’s needs. Both technology and business stakeholders are closely involved in this step.

Pilot Δ The step usually involves building a Minimum Viable Product (MVP) for AI capabilities and proceeding further based on the learnings from the MVP. It is critical to ensure that the MVP solves the business problem rather than simply demonstrating technical feasibility. The basic level of AI capability monitoring and measurement is applied at this step. Pilots should be designed to get to an MVP that can test the hypothesis of the AI capability as quickly as possible.

Operationalize ⚒ At this stage, the AI capability is proven viable in the previous step and gets designed and implemented to allow full integration with the existing solution ecosystem and provides full-fledged support for required business scenarios. Operationalization is expensive in solution development and may lead to significant waste if the MVP does not validate the viability of integration with existing systems. The organization must also plan for the investments required to monitor, adjust, and retrain AI components continuously.

Scale ⚒ An AI capability is successful if it grows to support increased volumes of customers. In the case of AI, unlike conventional solutions, scaling introduces some unique challenges. So, with increasing processing volume, the actual business parameters, the data, and the AI model may grow out of alignment (something called “drift” in AI terminology). Progressive adjustment of the model, the learning algorithm, and data processing is often required to sustain the model through scaling and beyond. Additionally, as a result of scaling, some qualitatively new scenarios may emerge or be subsumed by the solution. This may sometimes lead to a significant update to the AI capability or even a drastic change to the ML model or the learning algorithm.

Govern ⚒ Enterprises are often presented with multiple opportunities to apply AI and can do it in different areas. Over time, an ecosystem of intelligent capabilities emerges across various solutions operating in the same big data pool that establishes potential interoperability to create higher-level enterprise value. This requires a governance process that spans multiple Solution Trains in the portfolio or even [multiple portfolios](#) in the enterprise. This governance process can be a part of the general portfolio and enterprise governance. Still, it must especially account for some areas that grow far more important with the advent of AI. Such areas include effective data management, privacy and security, computing power, the bias problem, and traceability. Ethical implementation of AI is a dominant topic in this domain and requires serious investment by organizations leveraging this technology.

Due to the inherent complexity of AI solutions and the lack of adequately established practices supporting AI development and operation, many organizations experience serious challenges driving their AI solutions through operationalization, scaling, and proper governance. This significantly reduces the number of initiatives that produce any business value. That is why an organization must establish a productive solution path that fosters rapid feedback and effectively uncovers and manages technology and business risks.

Summary

Succeeding with AI is critical to surviving and thriving in the Digital Age. SAFe provides various tools that enable the successful development and delivery of AI-powered solutions. By organizing the teams and trains to incorporate AI capabilities, using practical, strategically sound decisions, and proactively building up the necessary cloud and big data capabilities, enterprises create the foundation for a productive AI effort. It is essential that technology and business professionals collaboratively develop competency around AI and understand where Artificial Intelligence can help the organization seize critical business opportunities.

Learn More

[1] PwC. *The Macroeconomic Impact of Artificial Intelligence*, 2018.

<https://www.pwc.co.uk/economic-services/assets/macroeconomic-impact-of-ai-technical-report-feb-18.pdf>

[2] PwC. *AI Predictions*, 2021. <https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-predictions.html>

[3] Gartner. *The 4 Trends That Prevail on the Gartner Hype Cycle for AI*, 2021.

<https://www.gartner.com/en/articles/the-4-trends-that-prevail-on-the-gartner-hype-cycle-for-ai-2021>

[4] Google. *Using GPUs for Training Models in the Cloud*.

<https://cloud.google.com/ai-platform/training/docs/using-gpus>

[5] Lansiti, Marco and Karim R Lakhani. *Competing in the Age of AI: Strategy and Leadership When Algorithms and Networks Run the World*. Harvard Business School Publishing Corporation, 2020.

[6] OpenAI. ChatGPT, 2021. OpenAI API, accessed February 5, 2023.

Last update: 7 February 2023

images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Éa management methodology that helps ensure that the company focuses efforts on the same important issues throughout the organization

ÑJohn Doerr,
Measure What Matters

Adjust Cookie Settings

OKRs

Objectives and Key Results (OKRs) is a collaborative framework for establishing clear goals and measurable outcomes.

When used within SAFe, OKRs can help to support the [Core Values](#) of transparency and alignment between the Enterprise and Portfolio strategy and the work of the Agile Release Trains and Agile Teams to deliver on this strategy. Additionally, OKRs can be applied to measure organizational improvement activities, including the desired outcomes for the SAFe transformation.

While this article describes opportunities to apply OKRs within SAFe, their usage is optional, apart from the strong recommendation to use them to describe portfolio [Strategic Themes](#). If a decision is taken to apply OKRs, these must be applied within an environment conducive to their success. The benefits of OKRs will only be realized where planning and delivery are incremental, ensuring the ability to respond to the continuous feedback that OKRs provide. Enterprises still employing traditional development methods, with an upfront commitment to large batches of work, will often struggle to reap the benefits of OKRs. We will conclude this article with a special caveat where OKRs do not work as well as might have been anticipated.

Details

Since their invention by Andy Grove at Intel and the subsequent, well-publicized adoption at Google in 1999, many leading technology companies have embraced Objectives and Key Results (OKRs). They create aspirational goals that drive higher organizational performance and employee engagement [1].

The simplicity of OKRs goes a long way toward explaining their popularity. The “Objective” defines the business outcome you strive to achieve. The “Key Results” are the measurable success criteria used to track progress toward the objective. For each objective, there are typically between two to five key results [2]. An example of an OKR used to define a SAFe strategic theme is shown in Figure 1.

Figure 1. A Portfolio strategic theme defined as an OKR

Figure 1. A Portfolio strategic theme defined as in OKR format

Writing Well-formed OKRs

Well-written OKRs can effectively align individuals and teams to measurable outcomes, but poorly written OKRs may have the opposite effect. Common anti-patterns include describing business-as-usual activities in place of aspirational goals, focusing on outputs rather than outcomes, and defining key results that describe a list of tasks or deliverables.

In contrast, well-written Objectives have the following qualities:

Inspirational: Each objective should describe an important and worthwhile goal. The objectives should be ambitious and move people out of their comfort zone to strive to achieve more. In short, they should inspire people to act.

Clear and Memorable: Objectives should be written succinctly in clear, memorable terms. Furthermore, there should not be too many objectives; 3-5 is

typical. The objective itself should be qualitative and not include numbers. These will come later in the key results.

Committed or Aspirational: Each objective should be marked as either committed or aspirational. Committed objectives describe things that must be done, such as meeting changes in legislation, whereas aspirational objectives are things that we hope we can achieve. We strive to achieve 100% of our committed objectives, while a 60-70% achievement is more realistic for aspirational objectives.

Doing Work or Improving Work: An additional distinction that may be useful is recognizing which objectives are related to developing or incrementing the Solution and which are focused on improving processes. The former is measured with business-orientated key results, while the latter will likely use a combination of [SAFe flow metrics](#). (An example of an improvement OKR is provided in use case 3 below).

Each objective is typically accompanied by 2 Ð 5 key results. Well-written key results have the following qualities:

Value-based: Key results should describe desired outcomes rather than the activities that drive these outcomes. Getting this right can be challenging. To move from activities to outcomes, the following questions can be helpful:

- What is the desired impact of this activity?
- How will I know this activity had an impact?
- What is the measurable result I hope to achieve in doing this activity?

Measurable: Each key result should be measurable and accompanied by a target number. Ideally, across all the key results, a mixture of leading and lagging indicators should allow progress to be measured at different times.

Gradable: The key results should also be gradable. In other words, we should be able to measure how much of that outcome has been achieved. This contrasts with key results that have a simple yes or no response. Typical formats for expressing gradable key results include:

- Should increase from X to Y
- Should decrease from Y to X
- Should stay above X
- Should stay below Y

The example below illustrates how applying this guidance to create well-written OKRs helps to move the conversation from output-focused OKRs to outcome-focused OKRs:

Figure 2. Moving from output-focused to outcome-focused OKRs

Figure 2. Moving from output-focused to outcome-focused OKRs

Applying OKRs in SAFe

As mentioned previously, the application of OKRs in SAFe is optional. The following three use cases describe potential applications, although this list may not be exhaustive. Consider the benefits they might bring to your organization for each of these. Be mindful that too many use cases implemented at once may create overheads that outweigh the benefits.

The application of OKRs within SAFe falls into three main use cases, illustrated in Figure 3 below:

1. Enhancing strategic alignment across a SAFe portfolio
2. Defining business outcomes for epics and lean business cases
3. Setting improvement goals for the SAFe transformation

Figure 3. Potential applications of OKRs in SAFe

Figure 3. Potential applications of OKRs in SAFe

Each of these is described in more detail below.

1. Enhancing Strategic Alignment across a SAFe Portfolio

The first use case is to apply OKRs to help with strategic alignment. Strategic themes provide the requisite transparency to ensure this alignment across the value streams and ARTs within the portfolio. SAFe recommends using OKRs for defining strategic themes, as shown in Figure 1 above, as it provides an effective means to define, organize and communicate these critical differentiating business objectives.

Defining Value Stream and ART OKRs

Strategic themes often don't impact all value streams in a portfolio equally; the work to execute against a particular strategic theme will differ for each value stream. Therefore, it can be helpful to create specific OKRs for each value stream that align with the portfolio strategic themes. And further, for large value streams

that contain multiple ARTs, this process can be repeated to create a set of OKRs that define the goals for each specific ART. Figure 4 below shows how these different OKRs work together to provide alignment and transparency at multiple levels.

Figure 4. Enhancing alignment across a SAFe portfolio with Value Stream and ART OKRs

This approach also allows those at each level of the organization to see the direct impact of their work against the OKR key results. Contrast this with the situation where goals are typically defined only at the highest level within the organization. And in the same manner that portfolio strategic themes influence key decision-making, ART OKRs become a critical tool for Solution and Product Managers, and Solution and System Architects when developing their roadmaps and exploring and prioritizing new opportunities.

OKRs inform Value Stream and Solution KPIs

SAFe [Key Performance Indicators](#) (KPIs) provide a set of measures that help to evaluate how a value stream or ART is performing against its forecasted business

outcomes. OKRs play a crucial role in informing and defining these KPIs.

NOTE: The difference between OKRs and KPIs: Whereas OKRs define the specific objectives we are working towards to achieve future success, KPIs represent ongoing ÔhealthÕ metrics that can be used to measure overall portfolio and business performance.

Value stream KPIs will be influenced directly by the portfolio strategic themes, which set clear expectations around the key results that must be delivered or supported. (This process is discussed further in the Value Stream KPIs article.) However, as explained above, specific OKRs can also be defined for value streams and ARTs. In this situation, the associated key results can be used to define KPIs for a particular solution that the ART is developing.

The Fulfilment ART, from Figure 4, is developing a Ôdelivery notifications and communication platformÕ to meet the goals outlined in its ART-specific OKRs. The following five solution KPIs emerge by analyzing the key results of those OKRs.

- Cost per notification
- Time to communication
- Missed deliveries
- Error rates in notifications
- Customer engagement

These solution KPIs not only help to measure the overall business outcomes from the current work but will continue to persist and evolve, separate to additional epics that get defined. In this manner, they provide a higher-level view of whether our solutions achieve their expected business outcomes.

Measuring Progress against OKRs

OKRs provide a means to continually measure progress, allowing the organization to take the necessary corrective actions or amplify successes. Each key result is measurable and should be gradable on an implicit scale, typically a percentage. However, because changes in strategy take time to apply, a quarterly cadence for measuring and reporting against key results is recommended. The [Strategic Portfolio Review](#) is the ideal event to reflect on progress. Figure 5 shows a set of key results measured quarterly.

Figure 5. Key Results measured quarterly

Figure 5. Key Results measured quarterly

The benefits of having multiple key results attached to each objective should be clear from this example. No single key result can provide the complete picture in isolation. Some improve based on the work done, while others go in the wrong direction. These multiple perspectives provide the inputs required to reflect on necessary changes to the plan.

Of course, getting feedback in this manner requires two critical things. Firstly, the ability to deliver value often and thereby incrementally measure the progress made. This requires investment in the [Continuous Delivery Pipeline](#) and the ability to continuously deploy and release value on demand. Secondly, the ability to measure must be built into the solution. For each key result defined, the question should be asked, "how are we going to measure this?" In many instances, this will require solution telemetry, which must be part of the solution development activities.

2. Defining Business Outcomes for Epics and Lean Business Cases

The second use case is to apply OKrs to help to uncover potential [Epics](#) for entry into the [Portfolio Kanban](#) system, as shown in Figure 6. The question that arises is, "What work do we need to do to affect a positive change in the key results we have defined?" This ensures alignment amongst those contributing ideas and an objective means of either approving or removing ideas from the funnel. Furthermore, for the epics that proceed to implementation, the connection to the strategy is clear for all working on them.

Figure 6. Epic ideas aligned to Strategic Themes defined as OKRs [3]

Figure 6. Epic ideas aligned to Strategic Themes defined as OKRs [3]

NOTE: Defining strategic themes as outcome-based OKRs in isolation from defining the Epics that will achieve them can often lead to a lack of feasibility in the strategy itself. It is all too easy for aspirations to become disconnected from the capacity and capability of the organization. To prevent this, it is recommended to combine the activity of defining the OKRs with a consideration of the potential epics that might execute that strategy. After all, outcomes cannot be achieved without outputs. Outputs have to come first, and they go hand in hand.

Epic and Lean Business Case Definition

OKRs can also be essential in epic definition, specifically when clarifying the desired business outcomes. The business outcomes are those measurable benefits that the business can anticipate if the epic hypothesis is correct. Applying OKRs to describe business outcomes ensures that they are outcome-focused and quantifiable. An example is shown in Figure 7 below. OKRs used in this manner as part of an Epic Hypothesis Statement or Lean Business Case may also inform portfolio prioritization conversations.

Figure 7. Using OKRs to define business outcomes as part of an epic hypothesis statement [3]

Defining an MVP

The next step in the lifecycle of an epic is to move it through the SAFe Lean Startup Cycle. Here the goal is to define a Minimal Viable Product (MVP) used to prove or disprove the epic hypothesis as quickly as possible. Therefore, the outcomes that this MVP is testing need to be clearly defined and measurable for us to make an

appropriate pivot or persevere decision. Once again, OKRs can provide this clarity. The objective defines the expected outcome of an MVP, and the key results represent the conditions that would prove the objective to be true, as shown in Figure 8.

Figure 8. Using OKRs to define an epic MVP [3]

Furthermore, once defined in this way, these OKRs—and those that describe the broader business outcomes—continually inform product management decisions made throughout the entire product lifecycle.

3. Setting Improvement Goals for the SAFe Transformation

The third use case is to apply OKRs to establish outcome objectives for the SAFe transformation itself. This can even establish a hierarchy of outcomes, as shown in the figure below:

Figure 9. Hierarchy of OKRs with examples

Figure 9. Hierarchy of OKRs with examples

In this scenario, the OKRs will predominantly fall into the "Improving work" category, typically focusing on improving quality, time to market, and predictability. An example of a single improvement objective and key results is shown below.



Figure 10. An example of a transformation improvement OKR

Applying OKRs to the SAFe transformation also forces early discussion around expected benefits. This generates the alignment and transparency previously described, and it also creates commitment to the journey that is about to begin.

A Special Caveat: OKRs and PI Objectives

We've described three use cases where OKRs can be beneficially applied in a SAFe implementation. And there are many more possibilities. But it's important to note a specific caveat concerning using OKRs to establish PI Objectives. The standard format for describing [PI Objectives](#) is to make them SMART. On the surface, it would seem a simple endeavor to change this to the OKR format. However, there are reasons we *caution against this*.

1. We must be mindful of the time it would take to write, review, and eventually assess the team PI objectives in the OKRs format. SMART PI objectives can typically be expressed readily in a single sentence or phrase. OKRs are much deeper and take

more time to discuss and write. Teams have limited time during PI Planning and drafting OKRs, each with 3-5 key results, leaving less time for more value-added discussions and planning activities.

2. Furthermore, key results often represent lagging indicators and often are not achievable within the PI timebox. This can be a source of frustration while establishing business value at the end of the PI.

So again, while this might be an obvious case for using OKRs, experience has shown them to be the wrong tool for writing PI Objectives.

Summary

The benefits of OKRs in creating alignment and supporting transparency are well documented, and within SAFe, they play a crucial role in defining strategic themes. Over time SAFe enterprises have also seen benefits in applying them in the other scenarios described above. When deciding whether to implement them in these additional ways, the expected benefits need to be carefully considered against the additional overhead in creating and maintaining them.

Finally, it should be recognized that OKRs can also fall victim to the same problems as the more traditional ways of defining outcomes. An organization that struggles to move the conversation to focus on business outcomes will only realize the benefits of OKRs if they ensure they are well-constructed and can measure progress against them on an ongoing basis. However, when done correctly, OKRs provide an effective tool to help drive better outcomes for the business and customers.

Learn More

[1] Doer, John. *Measure What Matters*.

[2] Castro, Filipe. *The Beginner's Guide to OKRs*.

Last update: 15 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK	PARTNER	CONTENT & TRADEMARKS	SCALED AGILE, INC
Download SAFe Posters & Graphics	Becoming a Partner Partner Directory	FAQs on how to use SAFe content and trademarks Permissions Form Usage and Permissions	CONTACT US 5400 Airport Blvd., Suite 300 Boulder, CO 80301 USA
Watch and download SAFe videos and presentations	GET SOCIAL		BUSINESS HOURS
Blog	Twitter Linkedin YouTube		Weekdays: 9am to 5pm Weekends: CLOSED
TRAINING			Privacy Policy Cookie Policy Your California Consumer Rights
Course Calendar About Certification Become a Trainer			

English

Scaled Agile Framework



Just like oil was a natural resource powering the last industrial revolution, data is going to be the natural resource for this industrial revolution.

ÑAbhishek Mehta,
CEO Tresata

Adjust Cookie Settings

Big Data

Big Data refers to the roles and practices required to collect, manage, normalize and deliver large datasets that help enterprises make more informed, fact-based decisions.

Data has become critically important across the entire enterprise. It influences business decisions, helps create better products, improves product development, and drives operational efficiencies. This article describes data's critical role in the enterprise, the *DataOps* process to manage and deliver extensive volumes of data, and how to apply DataOps in SAFe.

Details

In the digital age, enterprises generate data at an astonishing rate. Each website click, turbine engine rotation, vehicle acceleration, and credit card transaction creates new information about products, consumers, and operating environments. The rapid acceleration of information has led to new practices for storing, managing and serving massive data collections [1]. These Big Data practices deliver purpose-built data products to provide value across the entire enterprise, as Figure 1 illustrates.

Figure 1. Big Data products support all parts of the enterprise

Figure 1. Big Data products support all parts of the enterprise

The Evolving Role of Big Data in the Enterprise

Data accumulation typically begins within organizational silos. A department collects information about its users and systems to enhance products, discover operational improvements, improve marketing and sales, etc. While this localized data is valuable, aggregating large data sets across the entire organization provides exponentially more value than siloed data.

Exploiting Big Data for Competitive Advantage

Every enterprise uses data to improve its products, optimize operations, and better understand its customers and markets. Media and consumer product organizations use big data solutions to build predictive models for new products and services to anticipate customer demand. Manufacturing uses big data solutions for predictive maintenance to anticipate failures. Retail businesses utilize big data solutions to improve *Customer* experiences and effectively manage supply

chains. Financial organizations use big data solutions to look for patterns in data that indicate potential fraud.

Supporting AI Initiatives

Organizations use [Artificial Intelligence](#) (AI) and machine learning (ML) as a competitive advantage to provide better products to their customers, improve operational and development efficiencies, and provide insights that will enhance the business. Artificial Intelligence initiatives focused on machine learning require large sets of rich data to train and validate models. Lack of sufficient data is a common reason for the failure of AI initiatives. To achieve AI goals, an organization must develop an enterprise-wide approach to collecting, managing, and delivering data collected across the organization integrated with external data to fill gaps.

Big Data Challenges

Collecting and aggregating this data poses challenges. The data community characterizes Big Data with the Ô3 VsÕ:

Volume □ Data insights require a broad spectrum of data collected across the enterprise that can scale to hundreds of petabytes. As an example, Google processes 20 petabytes of web data each day. Big data solutions must collect, aggregate, and deliver massive volumes of data to data consumers.

Velocity □ Data-driven decisions require the latest data. Velocity determines how quickly new data is received and refreshed from data sources. For example, a Boeing 737 engine generates 20 terabytes of information every hour. Big data solutions must decide which data to store and for what duration.

Variety □ Data originates in many forms across the organization. Traditional data from databases, spreadsheets, and text are easy to store and analyze. Unstructured data from video, images, and sensors presents new challenges. Big data solutions must address all types of data.

More recently, the data community has added Variability, Veracity, Value, Visibility, and other ÔVsÕ to characterize Big Data further and add to the challenges of storing, managing, and serving it.

Understand DataOps in the Enterprise

To address these challenges, organizations need a unifying approach. The Data Science community recognizes organizationsÕ stages in the [Data Science Hierarchy of Needs](#) [2] (Figure 2). At the foundation, Data Engineers (likely [System](#) or [Solution Architects](#) on a [Development Value Stream](#)) design solutions that collect and manage data. This data and its storage are optimized for the application

without concern for broader use.

As this data becomes more pertinent to the organization, Data Engineers (usually as part of a centralized data function) transform and aggregate the broader sets of application data into a data warehouse to make it available through data products such as marts, cubes, and views. Data Analysts and others use these read-only, purpose-fit data products for statistical analysis and visualizations. Data Scientists use them to develop and train models for AI and ML.

Figure 2. The Data Science Hierarchy of Needs

Figure 2. The Data Science Hierarchy of Needs

Individuals often play multiple roles. For example, a Data Analyst creating a dashboard with Data Engineering skills may go back into the data warehouse to transform and re-aggregate data for a new or updated view. However, an organization's governance rules may restrict the ability of individuals to act in multiple parts of the pyramid. While most organizations take a centralized Data Warehouse approach to data, distributed strategies like the Data Mesh are emerging, particularly for large organizations.

The DataOps Lifecycle

The Big Data practices above are performed continuously as part of the *DataOps* lifecycle model shown in Figure 3. DataOps is a collaborative data management activity across [Agile Teams](#), data practitioners, and enterprise stakeholders that leverages Lean-Agile and [DevOps mindset, principles](#), and practices to deliver quality data products predictably and reliably.

Figure 3. The DataOps lifecycle

Figure 3. The DataOps lifecycle

The top portion of the lifecycle shows how siloed, application-specific data flows into data products offered to various consumers. The bottom part shows how data is consumed, governed, and used to enhance solutions that generate even more data into the pipeline. The remainder of this section describes each DataOps activity.

Collect Data ☰ System and Solution Architects, Agile teams, and system administrators build telemetry, logging, and monitoring to gather data on system

and user behaviors. [Product Management](#) prioritizes this work. System Architects ensure the solutions are easy to instrument and architected so the application data can be accessed externally, often through APIs. As data becomes more critical to the broader enterprise, work to collect better data and expose it through APIs may increase in priority and require capacity allocation (see [ART Backlog](#)) to ensure proper balance with other backlog items.

Aggregate & Transform Data Engineers transform and aggregate data from across the enterprise into normalized forms optimized for efficient use by data consumers. A centralized data architecture ensures efficient storage and delivery of consistent data products across the enterprise. Data Engineers must balance the $\hat{O}Vs$ discussed earlier and determine which data to store, how long, tolerable access times, etc. They view data as a product and apply [Design-Thinking](#) and [Built-in Quality](#). Personas and Journey Maps help them empathize with data consumers' pains, gains, and user experiences and determine how to offer better data products.

Deploy & Monitor Data Engineers deploy data products from the data warehouse into various forms, including data marts, cubes, and views used by data consumers. Like other technology solutions, data solutions leverage Cloud technology and apply [Continuous Delivery](#) through a DevOps pipeline designed for data. These practices quickly move data changes through development, Q/A, UAT, and production environments for consumer feedback. Multiple environments ensure dashboards, reports, models, and other artifacts dependent on the data content and formats can evolve with the data.

In the spirit of DevOps, monitoring occurs across all stages of the data pipeline to detect anomalies in the data (row counts, data out of range) and data operations (unexpected timeouts) and sends alerts to the data team.

Consume Big Data consumers can be categorized into two groups, as shown in Figure 4. Analysts use data products to discover insights and create visualizations for specific data customers. Data Scientists and ML developers use them to develop and train models. Their customers (Data Customers in Figure 4) are enterprise-wide stakeholders seeking business insights for decision-making, improving operations, and enhancing solutions.

Figure 4. Big Data has many customers across the entire enterprise

Figure 4. Big Data has many customers across the entire enterprise

Different lines of business, product managers, AI developers, and others have unique data needs. To enable these individuals through a self-serve model, enterprises should invest in the tools, analytics packages, and resources to support ad hoc access for localized manipulation. This investment reduces the load on members of the centralized data team for reporting and other mundane tasks.

Govern ☐ DataOps must enforce data privacy, confidentiality, residency, sharing, retention, and other legal requirements. The Big Data solution must ensure security through access controls, audits, and monitoring that detect intrusion and data breaches. Like other digital products, it must also provide fault tolerance and disaster recovery through vendors or home-grown solutions.

Enhance Solutions ☐ Portfolio and [Agile Release Train](#) (ART) leaders use data to enhance solutions that provide better customer value (better products) and business value (better data). Data analytics can reveal opportunities for new solution offerings and inform feature prioritization for existing solutions. Data gaps also show opportunities to enhance solutions to collect additional data.

Applying DataOps in SAFe

The previous sections describe the importance of a clear and compelling data strategy. This section describes additional guidance SAFe organizations can take to support their Big Data journey.

DataOps is a Portfolio Level Concern

SAFe addresses Big Data concerns at the portfolio level as it requires vision, investment, and governance at the highest levels in the organization (Figure 5). While ARTs create the data, the value comes from data aggregation at the portfolio and enterprise levels. Big Data solutions require strategic investment from the organization and a comprehensive approach that aligns each of the organization's development value streams to common DataOps practices that produce cohesive data sets used across the entire organization. Portfolio leaders use [Lean Budgets](#) to invest in Big Data infrastructure and DataOps practices to accomplish this. They also use [Portfolio Epics](#) and the [Portfolio Backlog](#) to specify and prioritize the infrastructure, technologies, and data needs to support the organization (Figure 5).

Figure 5. Big Data is a portfolio concern

Figure 5. Big Data is a portfolio concern

Initially Centralize the Data Function

SAFe's [Principle #10, Organize Around Value](#), strives to optimize flow by ensuring teams and ARTs have all the skills necessary to deliver value. Unfortunately, most organizations are still growing their data engineering and science functions, resulting in more demand than capacity for these skills. Initial centralization is often helpful for early technology adoption to maximize available skills and create the DataOps infrastructure and practices. Centralization also simplifies governance for privacy and security that would be virtually impossible to safeguard with a siloed approach.

This centralized function can meet most of the organization's data needs through the customer-centric approaches described earlier. Where additional support is needed, SAFe has a known pattern of providing additional services to other parts of the organization through a [Shared Service](#), as shown in Figure 6.

Figure 6. Start with a centralized data function that supports other parts of the organization as a Shared Service

Over time, organizations will grow their data functions to support the broader enterprise and embed individuals in ARTs and [Operational Value Streams](#). However, a ÔBig DataÕ development value stream of Data Engineers providing aggregated enterprise data will likely exist for some time.

Grow Technical Talent

Data Engineering, Analysts, and AI/ML developer skills are in huge demand, and recruiting knowledgeable and skilled data specialists is a significant challenge. Organizations must create a compelling and inspiring data and AI vision to attract and retain this talent. Data specialists want to learn and grow from other data specialists to keep pace with the rapidly evolving technologies and practices.

Applying DataOps to Build Better Solutions

Solutions are informed and augmented by data, and ARTs require Data Science and Data Engineering skills. As described earlier, the data functions can provide some resources to ARTs as a shared service. But they must balance that ART work with their primary responsibility to create and evolve the DataOps practices that include providing data products to the enterprise. When supporting ARTs, they should act like an Enabling Team (see [Agile Teams](#)) to grow the technical data competence across the organization. In this capacity, they are not there to do the work but to teach others how to do it.

Learn More

[1] Kleppman, Martin, *Designing Data-Intensive Applications*. O'Reilly Media, 2017.

[2] The data science pyramid. <https://towardsdatascience.com/the-data-science-pyramid-8a018013c490>

Last Update: 9 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Linkedin](#)

[YouTube](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The Ôrelay raceÕ approach to product developmentÉ may conflict with the goals of maximum speed and flexibility. Instead, a holistic or ÔrugbyÕ approachÑ where a team tries to go the distance as a unit, passing the ball back and forthÑ may better serve todayÔs competitive requirements.

[Adjust Cookie Settings](#)

ÑHirotaka Takeuchi

and Ikujiro Nonaka,
ÒThe New New
Product
Development
GameÓ [1]

SAFe Scrum

Find a Course:

Note: For more on SAFe Scrum, please read the additional Framework articles in the Scrum series, including [Scrum Master/Team Coach](#), [Iterations](#), [Iteration Planning](#), [Iteration Goals](#), [Iteration Review](#), and [Iteration Retrospective](#)

Go

SAFe Scrum is an Agile method used by teams within an ART to deliver customer value in a short time box. SAFe Scrum teams use iterations, Kanban systems, and Scrum events to plan, execute, demonstrate, and retrospect their work.

Many teams use SAFe Scrum as their primary Agile Team process. However, they are not limited to Scrum and may also use [SAFe Team Kanban](#). However, all SAFe teams apply [Built-in Quality](#) techniques and other practices and activities not defined in Scrum. SAFe Scrum teams cooperate with other Agile Teams and ART stakeholders, building and deploying [Solutions](#) that benefit its Customers.

This article describes how Agile Teams apply SAFe Scrum.

Details

Agile Teams applying SAFe Scrum follow a regular cadence of events to achieve a common objective, delivering value to the enterprise and its customers. Teams have the authority and autonomy to plan, execute and manage their work, make decisions within their scope, and adapt to changing conditions in the best way they see fit. SAFe [Principle #9, Decentralize decision-making](#), ensures alignment, and fosters subtle and minimal management direction for SAFe Scrum teams.

Indeed, teams determine how they do their work and the scope they can commit to within the iteration time box. They create and refine backlog items, expressed as [Stories](#) and acceptance criteria, defining and committing to iteration goals. They then build, test and deploy the new functionality, and ensure built-in quality for each solution increment.

Moreover, since Scrum teams have all the roles and skills needed to develop and deliver increments of value, they are designed to operate with the minimum possible constraints and dependencies with other teams. The self-management and cross-functional nature of the Scrum team—along with constant communication, constructive conflict, and dynamic interaction — creates a more enjoyable, fun, and productive work environment.

The SAFe Scrum Cycle

Figure 1 illustrates the basic Scrum cycle using SAFe terminology (e.g. iteration vs. sprint). Each event within this cycle provides an opportunity to inspect progress and make mid-course corrections. These events are often held at the same time and place to reduce overhead.

Figure 1. SAFe Scrum cycle

Figure 1. SAFe Scrum cycle

The following sections provide an overview of this cycle.

Scrum [Iterations](#) and events, including [Iteration Planning](#), [Review](#), and [Retrospective](#), are briefly described in the following sections and in more detail in SAFe articles.

Team Backlog

The [Team Backlog](#) holds all the upcoming work needed to advance the solution. Most of the work is captured by user stories, but other work and activities (e.g., training, support) may also be included. The team backlog will have been partially filled during [PI Planning](#), and the teams will have established their [PI Objectives](#). Teams continually refine the backlog to ensure it always contains some stories ready for implementation without significant risk or surprise. During backlog refinement, teams review upcoming user stories and [Features](#) (as appropriate) to define, discuss, estimate, and establish acceptance criteria for upcoming backlog items. Backlog refinement is a continuous process throughout the iteration.

Iteration

[Iterations](#) (Sprints) are the heartbeat of Scrum and create a rhythm for work within the larger PI timebox. All the work necessary to achieve the iteration—including planning, team sync, iteration review, and retrospective—occur within iterations. Each iteration is a standard, fixed-length timebox, typically two weeks in length.

The team delivers high-quality increments of value during the iteration as working, tested software, solutions, or other valuable artifacts. Iterations are continuous and sequential, and a new iteration starts immediately after the previous one.

Iteration Planning

[Iteration Planning](#) is the first event of the iteration. The [Scrum Master/Team Coach](#) typically facilitates this event. All team members collaborate to determine how much of the team backlog they can deliver during an upcoming iteration and summarize those stories into a set of iteration goals. The team records the resulting plan in the iteration backlog. The [Product Owner](#) ensures that attendees are prepared to discuss the most critical team backlog items and how they map to the iteration goals and PI objectives. The Scrum team may invite other people to plan and provide advice.

Iteration planning addresses the following topics [2]:

- Why is this iteration valuable?
- What can be done in this iteration?
- How will the work get done?
- How will the chosen work get done?

Throughout iteration planning, the team elaborates the acceptance criteria for each story and estimates the effort to complete it. The team selects candidate stories based on their available capacity for the iteration. For each selected item, the team plans the work necessary to create an increment of value that meets the definition of done (DoD), ensuring that it can be considered complete. The continuous development of incremental system functionality requires a scaled definition of done to ensure that work is done right (see the Scalable definition of done in the [Built-in Quality](#) article).

Iteration planning often requires decomposing backlog items into user or [Enabler](#) stories that can typically be completed in a day or two. The team decides how the work gets done within the iteration. Once planning is complete, the team commits to the work and records the iteration backlog in a visible place, such as a storyboard, Kanban board, or Agile project management tooling. Planning is timeboxed to a maximum of about two hours for a two-week iteration.

Iteration Goals

Iteration goals are a high-level summary of the business and technical goals that an Agile Team agrees to accomplish in an iteration. They are vital to coordinating an ART as a self-organizing, self-managing team of Agile teams. Iteration goals provide the following benefits:

- Aligns team members to a common purpose
- Aligns teams to common ART PI Objectives and helps manage dependencies
- Provides transparency and management information

Once the team completes iteration planning and creates the iteration backlog, the team synthesizes the work for the next iteration as a set of iteration goals, which defines why the iteration is valuable. These goals are derived from the iteration backlog and aligned with the PI objectives. Optionally, the PO may define an initial set of goals before planning; this helps set the stage for the “why” and “what” of planning.

Iteration goals provide Agile teams, ART stakeholders, and management with information and a shared language for maintaining alignment, managing dependencies, and making necessary adjustments during the execution of the PI.

Iteration Backlog

During iteration planning, teams pull items from the team backlog to create the iteration backlog, containing the things they intend to complete in that iteration. Since PI planning is high-level, adjustments will likely need to be made for each iteration. Moreover, the teams get feedback—not only from their prior iterations but also from the [System Demo](#) and other teams with whom they collaborate. The local team’s concerns (defects, tech debt, and maintenance), ART backlog, and committed team PI objectives all influence the content of the iteration backlog.

Deliver

In SAFe, Scrum teams operate within the cadence and synchronization requirements of the ART, facilitating alignment, dependency management, and fast integrated learning cycles for the entire Solution.

During the iteration, each team collaborates to define, build, and test the stories they identified during iteration planning. They track the iteration’s progress and improve the flow of value by using story and Kanban boards and team sync events. They deliver stories throughout the iteration and avoid “waterfalling” the timebox. They apply Built-In Quality practices to build the system right. These completed

stories are demoed throughout the iteration and at the Iteration Review.

Team Sync

The team sync is a short meeting (usually 15 minutes or less), typically held about daily, to inspect progress toward the iteration goal, communicate, and adjust upcoming planned work. The team can use any structure or technique for the team sync, creating an action plan for coordinating the next workday. But the team sync is not the only time team members can make adjustments. The team collaborates and discusses adapting or re-planning work whenever needed.

High-performing teams use the team sync to find opportunities to help each other so that the entire team succeeds in delivering its committed iteration goals. The Scrum Master/Team Coach writes down topics that need further discussion on the Ômeet afterÕ board. During the meet after, only the involved parties stay to discuss these topics in more detail. Ineffective team syncs may be a symptom of deeper problems that require a systemic approach for resolution, which often becomes the responsibility of the Scrum Master/Team Coach.

Increment

The increment illustrated in Figure 1 represents how new solution functionality evolves within each iteration. Each increment is additive to all prior ones and verified, ensuring they all work together. Each is a working, tested and usable solution that meets the agreed quality criteria in the definition of done (DoD). Further, the team can deliver multiple increments within an iteration. The sum of all the teamÕs increments is inspected at the iteration review.

Iteration Review

For Scrum Teams, the [Iteration Review](#) is the second to last event of the iteration, where each team inspects the outcomes of the iteration, presents the results of their work to key stakeholders, and assesses progress toward the iteration goal and PI objectives. It also determines any future adaptations to the product or process. Based on the results, stakeholders and the team collaborate on what to do next. The [Team Backlog](#) may also be adjusted to meet new opportunities. The iteration review is timeboxed to a maximum of two hours for a two-week iteration.

Iteration Retrospective

The [Iteration Retrospective](#) is the last event of the iteration. Its primary purpose is to reflect on the iteration and derive new ideas to improve its process and the solution. The retrospective helps instill the concept of relentless improvement—none of the SAFe [Core Values](#). The team discusses what went well, what problems it

encountered, and how those problems were (or were not) solved. The team identifies the most helpful changes to improve. These changes are addressed as soon as possible or recorded in the backlog for the next iteration. The retrospective is timeboxed to a maximum of about 60 minutes for a two-week iteration.

Participating in the System Demo

The [System Demo](#) provides an integrated view of new features for the most recent iteration delivered by all the teams in the ART. Each demo gives ART stakeholders an objective measure of progress during the PI. Everyone on the ART participates or at least attends the system demo.

Tracking Progress

The team tracks its iteration progress using a Kanban board (Figure 2). Mature Scrum teams deliver stories throughout the iteration without “waterfalling” the timebox.

Figure 2. An example of a team backlog and workflow represented on a Kanban board

SAFe Scrum Roles

The Scrum team’s size (typically 10 or fewer) and structure optimize communication, interaction, and the ability to deliver value. The Scrum team is small enough to remain nimble and large enough to complete significant work within a sprint [1].

All SAFe Agile Teams, including those running Scrum, have two specialty team roles, each with a unique set of responsibilities (Figure 3):

The [Scrum Master/Team Coach](#) (which may be a part-time role) is responsible for helping coordinate the flow of work to deliver to customers. They foster an environment for high performance and relentless improvement by coaching Scrum and Lean-Agile principles, helping remove impediments to progress, and facilitating team self-organization and self-management. Many SAFe Scrum Masters/Team Coaches have additional coaching responsibilities, such as coaching DevOps, Built-in Quality, Kanban, and flow.

The [Product Owner](#) (typically a full-time role) understands the needs and expectations of customers. They serve as the team backlog content owner and prioritize the team backlog, helping to ensure the team is building the right thing in the right way.

Figure 3. The SAFe Agile team

Figure 3. The SAFe Agile team

Agile Teams are on the Train

Although teams are cross-functional, a single Agile Team may not be able to deliver complete end-user value for large systems that may include different technology platforms and a spectrum of disciplines such as hardware, software, and systems engineering. However, to optimize flow, teams should strive to deliver end-to-end value as independently as possible. In support of the larger purpose, SAFe Agile teams operate within an [Agile Release Train \(ART\)](#), providing an environment for aligning and collaborating with other teams to build larger solutions and figure out how to reduce or eliminate dependencies and improve [Team](#) and [ART Flow](#).

As part of the ART, all teams plan, demo, and learn together, as illustrated in Figure 4, which helps them focus on both local concerns and the larger aim of the ART. This alignment also enables teams to explore, integrate, deploy, and release value more independently.

Figure 4. Agile Teams plan, demo, and learn together

Figure 4. Agile Teams plan, demo, and learn together

The [Agile Teams](#) article further defines each team's participation in this shared responsibility of delivering customer value.

Learn More

[1] Takeuchi, Hirotaka, and Ikujiro Nonaka. *“The New, New Product Development Game,”* Harvard Business Review, January 1986.

[2] Sutherland, Jeff, and Ken Schwaber. *“The 2020 Scrum Guide,”* Scrumguides.org.

Last Update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Tuck your team members in around the areas where they quickly achieve flow—those are typically where they are particularly primed to contribute value.

—Brené Brown, *Dare to Lead*

Team Flow

[Adjust Cookie Settings](#)

Team Flow describes a state in which Agile teams deliver a continuous flow of value to the customer.

The SAFe [Team and Technical Agility](#) competency guides the creation of effective cross-functional Agile Teams and Agile Release Trains (ARTs). It encourages teams to apply specific quality practices and work with extended stakeholders to deliver valuable products and services to their customers. This competency has proven to be highly effective in improving business outcomes. But many individuals and teams have not worked in an Agile way before, and impediments abound. Moreover, there is no limit to a team's effectiveness, and there are always opportunities to improve the flow of value to the customer.

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of eight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Details

Agile Teams do much of the vital work of defining, building, validating, deploying, and supporting some of the world's most important systems. The small, cross-functional nature of Agile teams, the increased level of empowerment, the connection to the customer, and proven Agile practices all contribute to creating a work environment that delivers faster value, is far more productive and is simply more fun.

However, the work itself is complex, and while that is a big part of what brings knowledge workers to the endeavor, it also complicates the effort. SAFe addresses this challenge by providing guidance and practices for the roles and activities of

each Agile Team. When implemented correctly, work will flow to the customer as smoothly as the process allows. Much of this guidance is provided in [Principle 6 – Make value flow without interruptions](#), which highlights eight properties of a system of value flow. Each of the eight properties offers an opportunity for inspection and analysis to determine what interruptions to flow are likely to occur. While the properties and accelerators are common to all flow systems, the application is specific to each level of SAFe. The sections below describe how to apply the accelerators to Team Flow.

#1 Visualize and Limit WIP

Why it matters

Excessive Work in Process (WIP) decreases team productivity and impedes the flow of value to the customer (Figure 1). It confuses individual and team priorities, causes frequent context switching, and increases waste, overhead, and frustration. Like a highway at rush hour, there is simply no upside to having more work in a system than the system can handle.

Figure 1. Excessive WIP slows down value delivery

Figure 1. Excessive WIP slows down value delivery

What to do about it

Make the current WIP visible. The first step is to make all the different types of work visible, including new functionality, maintenance, architecture, infrastructure, and technical debt reduction. Often, this simple visualization is a much-needed Ôwake-up callÕ that causes practitioners to start addressing the systemic problems of too much work and too little flow.

Balance WIP against available capacity. The next step is to set WIP limits that balance demand against the available development capacity. No new work is started when any workflow state reaches its WIP limit, helping match demand and capacity and increasing flow.

#2 Address Bottlenecks

Why it matters

A teamÕs productivity is constrained by various bottlenecks Ð where skill, capacity, resources, and other elements of the process or system cannot meet the demand.

Improving team flow requires addressing bottlenecks continuously.

What to do about it

The first important task is to identify bottlenecks. Common examples include:

- An insufficient number of people with a given expertise
- Overspecialization
- Excessive technical debt
- Lack of availability of a shared service
- Lack of customer feedback

Many bottlenecks occur when work passes from one workflow step to another. A large amount of work piling up in front of a state in the Team Kanban may indicate a bottleneck (Figure 2).

Figure 2. Work piles up in front of the bottleneck

Figure 2. Work piles up in front of the bottleneck

However, when bottlenecks are caused by a capacity or skill shortage, identifying them is not always obvious and may require root cause analysis.

Once discovered, there are generally **two** ways to deal with a team bottleneck.

1. Increase capacity at the bottleneck. In many cases, adding people is an appropriate response. But there are other ways as well. SAFe's [Built-in-quality](#) and other Agile practices provide many helpful techniques for increasing capacity at the bottleneck, including:

Collective ownership. Collective ownership implies that anyone on the team can fix a defect, update a work product or resolve a problem. However, to help ensure that there are no unintended consequences, such changes are subject to the necessary competence and local governance of the work product.

Developing T-shaped skills. Agile Teams develop people with T-shaped skills. This metaphor describes people with deep skills in a specialty area and broad—but not necessarily deep—skills in other areas. T-skills increase the range of issues and opportunities a team member can address.

Pairing. Pair work doubles the capacity to address the issue and provides better insights into a specific problem that may be blocking progress. It also helps in developing “T” skills.

Swarming. With “swarming,” multiple team members collaborate to accomplish work that a team member may have difficulty completing on their own.

2. Bypass the bottleneck. Bottlenecks are not always easy to resolve immediately at the source. Fortunately, teams likely have other valuable stories in their backlog that other non-constrained people can work on. Selective replanning can increase flow while the bottleneck is being addressed.

It's also important to note that some root causes of team bottlenecks may be beyond the team's span of control. In this case, the [Inspect & Adapt](#) event is an excellent opportunity to elevate the problem.

#3 Minimize Handoffs and Dependencies

Why it matters

“Handoffs” occur when work transitions from one process step to the next. “Dependencies” occur when specific input is needed from another person or a team. While some dependencies and handoffs are inevitable, excessive and unnecessary dependencies and handoffs disrupt team flow, create delays, and increase context-switching and overhead.

What to do about it

Organize around value. SAFe Principle 10 – *Organize around value*, provides the primary constructs for minimizing handoffs and dependencies. Effective team topologies help by reducing the cognitive load placed on the team.

Make handoffs and dependencies visible. Teams must understand what they are, the pattern in which they occur, and the impact they create.

Take corrective action. Corrective action may be evident from the context of the dependency. Such action often requires changes to the process, the design of the system under construction, the individual work product, or the organization or skills of the teams themselves.

#4 Get Faster Feedback

Why it matters

Solution development is critically dependent on the feedback needed to guide the team's work. When feedback is missing or delayed, misunderstandings accumulate quickly, leading to rework, slow delivery, and unsatisfied customers. Generally, two types of feedback are required, "building the right thing" and "building it right" as illustrated in Figure 3.

Figure 3. The PDCA cycle provides two different kinds of feedback

Figure 3. The PDCA cycle provides two different kinds of feedback

What to do about it?

Determine what types of feedback are missing or inadequate. Different forms of feedback provide distinct answers to various questions. Are the priorities right? Will the architecture support the functionality? Will the customer be able to use the product?

Shift reviews left. All work products should be peer-reviewed as early in the cycle as possible. However, there can be a natural tendency for individuals to want to get their work to a state of completion *before* showing it to others. But this can create over-investment in unproven designs and a defensive posture by the author. Apply transparency. Shift reviews for all WIP *left*.

Demonstrate working systems. SAFe has events for getting fast feedback. The most basic example is the [System Demo](#) that occurs every two weeks. It creates a discipline for validating core assumptions.

Frequently integrate and engage with the customer. Frequent integration provides critical feedback concerning the current implementation approach. Also, providing customer access to the current product increment in a staging environment can provide early important feedback.

#5 Work in Smaller Batches

Why it matters

Working in large batches of stories, tasks, and other activities leads to delayed feedback, significant rework, and high variability. Teams have many different types of batches in their flow system:

Integration batch (how much functionality can the team develop without integrating with others?)

Testing batch (how much functionality can the team create without systematic testing?)

Architectural batch (how much architectural enablement can be created without validating user-facing features?)

Customer feedback batch (how much functionality can be implemented without customer feedback?)

Deployment batch (how much functionality can be implemented without deploying and testing it in the production environment?)

What to do about it

Use recommended cadence and team size. SAFe structural guidance helps keep batch sizes small. Adhering to short PI and iteration lengths makes batch sizes smaller. In addition, the optimum size of ARTs and teams automatically imposes a limitation on how much work can be processed at a time.

Adjust the process to support smaller batches. If a specific batch remains too large, reducing it may require adjustments to planning and execution. It may also

require context-specific changes to reduce the effort in processing the batch.

Ensure proper enablement. Reducing batch size often creates a higher number of smaller transactions. Optimizing and reducing the batch size usually entails planning and executing enablers to refactor architecture and infrastructure

#6 Reduce Queue Lengths

Why it matters

Queues occur when there is a committed backlog awaiting implementation. The longer the queue, the longer the wait time for the new functionality.

What to do about it

Teams generally have some direct control over the length of each queue. They can make queues shorter with various mechanisms:

Flow measures like velocity and load (see below) help the team commit to only their available capacity. PI objectives help teams limit commitments beyond that scope, ensuring that the queue for new work is not much longer than a PI

Short iteration time boxes and concise iteration objectives also bring focus to near-term work

Informal or formal requests for additional work should be directed into the backlog to keep queues shorter

#7 Optimize Time ÐIn the ZoneÐ

Why it matters

Solution development relies heavily on creativity, focus, and the intellectual effort of team members. For example, implementing a new software feature requires a developer to navigate hundreds of intricate dependencies between different parts of the code in a way that is uniquely related to the component under development. The ÐzoneÐ is a highly productive mental state that takes time to enter and sustain.

It may take up to 15-20 minutes for a person to fully immerse themselves in the context of their work, and a simple external factor may instantly interrupt it.

What to do about it

Several factors make it difficult for individuals and teams to reach and sustain an optimal time in the zone. These include excessive WIP, excessive meetings, frequent context switching, interruptions, inadequate development tools, and CD

Pipeline infrastructure.

Ensuring that these factors are resolved is an essential task for every leader and coach. The following can help to increase time in the zone:

Optimize meetings and events. Teams should evaluate the efficiency of all work sessions. A team may discover, for example, that it remains more productive when the team members interact with each other informally throughout the day vs. a cadenced-based event (for example, the daily standup). Others find scheduling regular synchronization meetings add cadence to communication and minimizes interruptions. Teams should periodically review all their meetings to understand the necessary ones and the required attendance.

Limit work-in-process further. Context switching is often a result of too much WIP. Less active items at any given time mean fewer interruptions of work.

Use innovative collaboration patterns. Being in the zone does not imply working in isolation from others. Practices like pair work and mob programming can increase in-depth immersion and focus.

Improve work product health. Teams need to maintain a healthy code and asset base; otherwise, it takes more work to evolve and maintain the system efficiently.

#8 Remediate Legacy Policies and Practices

Why it matters

It's appropriate for those leading a transformation to want to leverage what's new. Far too often, however, leaders often wish to keep what's old. Some of these new and old practices are incompatible (design evolves vs. design upfront) and put teams in a difficult predicament. Either pretend to conform to mutually conflicting policies and reduce transparency or fight the system while development is in process, slowing progress and creating internal friction. In any case, they inhibit flow through the system and put the entire enterprise in half-Agile mode.

What to watch out for

The first thing is for leaders and change agents to recognize that legacy policies and practices are likely to persist— even with top leadership support for the Lean-Agile transformation. Some examples follow:

- Traditional project and program management (for example, Earned Value Management, Work Break Down Schedule, Integrated Master Schedule, and project cost accounting) on top of Agile team practices
- Keeping management status reporting in place when the system has moved to transparency and objective evidence
- Having developers maintain their old timesheet and recording work in their Agile

Lifecycle Management (ALM) tooling results in duplicate reporting
Requiring, for example, documenting design specifications, mandating traceability of non-critical code, and other practices not required by regulation because “we’ve always done it that way.”
Forcing root cause analysis and documentation on every defect, as opposed to finding, fixing, and forgetting
Separating developers and testers and V&V teams for “separation of quality concerns” when not mandated
Subjecting teams to legacy phase gate reviews
Required managerial review of artifacts and tasking micro-management
Compensation and performance policies that reward the wrong kind of behavior
Collecting and using metrics across teams as performance measures

What do about it

Unfortunately, this is nowhere near an exhaustive list. Leaders, Scrum Master, coaches, and SPCs should constantly look for these and other impediments to flow. The LACE should deal with known issues as part of the transformation roadmap. Newly discovered legacy practices and policies that cause impediments or extra work should be taken care of immediately.

Measuring Team Flow

It is hard to improve what isn’t measured. SAFe’s Measure and Grow guidance provides three measurement categories—Competency, Flow, and Outcomes—that can assess and improve an enterprise’s ability to deliver innovative business solutions quickly. This guidance includes six measures specific to flow: *flow distribution, velocity, time, load, efficiency, and predictability*. Flow velocity and distribution are particularly relevant to team flow and are highlighted below.

Flow Velocity

Figure 4 illustrates an example of a team’s *flow velocity*. In this example, the newly formed team has unstable velocity early on but sees some higher throughput. However, they later discovered that those stories didn’t integrate well into the broader system or perhaps did not meet all the NFRs. In other words, they weren’t indeed *done*.

The appropriate response is to tighten the Definition of Done (DoD) to a higher quality level. Doing so triggers lower delivered velocity, but each story has higher quality. Over time, teams sort it out and move to a more stable, higher quality, and higher-performing state. As noted earlier, quantitative data should be combined with qualitative data to interpret the information correctly.

Figure 4. An example of a team's flow velocity

Figure 4. An example of a team's flow velocity

For example, a velocity change can indicate an actual increase or decrease in a team's underlying productivity, and it usually is. But it could also reflect a move to a new business domain or technology stack or an effort to shift to smaller work items to improve flow.

Equally importantly, these measures *cannot* be used to compare Agile teams. Each team's local context can be different. Any attempt to set specific targets or use velocity as a performance measure will fail to cause improvement and may even have the opposite effect.

Flow Distribution

Figure 5 illustrates an example of a team's flow *distribution*, showing the various types of work items present in the system, as measured at each PI boundary.

Figure 5. One team's flow distribution over time

Figure 5. One team's flow distribution over time

Generally, the balance of work items is indicative of team and system health. In this case, it looks like a significant investment in architecture (enablers) was required to eliminate some tech debt or support some upcoming new kinds of functionality. However, these measures can also vary dramatically from team to team, based on each team's context and responsibilities. For example, some teams will have high maintenance responsibilities for current systems; others may be doing all new development.

In addition to velocity and load, the additional SAFe flow measures of *flow time*, *load*, *efficiency*, and *predictability* are equally valuable to the teams. As noted earlier, quantitative measures should be combined with other qualitative insights. Judgment is always required.

Last updated: 9 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



If we need synchronized efforts. Then the contribution of any single person to the organization's purpose is strongly dependent upon the performance of others.

—Eli Goldratt

ART Flow describes a state where an ART delivers a continuous flow of valuable features to the customer.

SAFe guides ARTs, working with their extended stakeholders, can get closer to their customers and build Continuous Delivery Pipelines (CDPs) to accelerate the delivery of valuable products and services. This guidance has proven highly effective in improving business outcomes for SAFe Enterprises.

But this type of digital transformation is complex, as any enterprise is complicated, and adopting a Lean-Agile way of working is a significant change. Many individuals and teams have not worked in an Agile way before, and impediments abound. In addition, there is no apparent limit to how effective an ART can be. There is always an opportunity to improve the flow of value delivery. Indeed, because of the scope and impact on customer value delivery, each ART may be the primary opportunity for improving business outcomes via improving continuous flow.

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of eight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Details

As highlighted in [Principle 6](#) [*Make value flow without interruptions*](#), SAFe defines a set of eight flow accelerators that can address, optimize, and debug issues to achieve a continuous flow of customer value. This article describes how to accelerate the flow of value through [Agile Release Trains](#) by applying the eight flow accelerators.

#1 Visualize and Limit WIP

Why it matters

Excessive Work In Progress (WIP) significantly decreases ART productivity and impedes value flow. It overloads people and teams, confuses priorities, causes frequent context switching, and creates long waits for new functionality.

An ART strives to do its best. However, it's normal for an ART to have lots of work in process (WIP), often far more than it can achieve over time. But that's counterproductive, as an overloaded ART will complete less work than it would otherwise.

What to do about it

Visualize all features in progress. An ART should manage the inventory of all the [Features](#) currently in flight. These are maintained and WIP-limited in the [ART Backlog](#) Kanban system.

Establish control with capacity allocation. Not all of the ART's work is expressed in features. Therefore, an ART's total WIP combines effort in current features and non-feature-related work: infrastructure, tooling, change programs, team and customer meetings, training, and more. It's easy to underestimate the time devoted to these other essential activities. To address the total WIP, establish capacity allocations for ART feature and other work and adjust over time.

#2 Address Bottlenecks

Why it matters

A bottleneck constrains the productivity of the entire ART, so teams must address them to improve flow. Once the current bottleneck has been resolved, the next one materializes and must then be taken care of to reach the next level of performance. It's a continuous process.

What to do about it

Identify the bottleneck. An ART can only address *known bottlenecks*. Some ART bottlenecks can be identified during [PI Planning](#); others will only materialize during PI execution or [Inspect and Adapt](#). Typical symptoms of a bottleneck may include:

- An overloaded individual team or a group of teams
- Repeated slippage of an activity (such as integration, testing, deployment, refactoring) from one iteration to another
- Delayed execution of certain types of dependencies

SAFe metrics and tools like Value Stream Mapping, the planning board, and the ART

Kanban system may help identify bottlenecks.

Understand the full impact. An ART needs to understand how its current bottlenecks affect the value flow. For example, slow customer feedback may lead to the ART building the wrong solution functionality, significant rework, and dissatisfaction. A deteriorating system architecture makes the development process significantly longer and less predictable.

Increase capacity at the bottleneck when possible. Once the problem has been identified and understood, increasing capacity at the bottleneck is an obvious remedy. For example, an ART that has an insufficient number of front-end developers may bring in more people. Or, in the case of poor architecture, the ART may allocate more time for tech-debt reduction.

Bypassing the bottleneck. However, increasing capacity at the chokepoint can sometimes be tricky as additional people and resources may not always be available. Near term, avoiding the blockage may be more productive. Selecting the next-most-valuable feature that does not have dependencies is one way to do this. For example, instead of launching a large-scale architectural improvement of a legacy system, an ART may choose to hasten the system's retirement.

#3 Minimize Handoffs and Dependencies

Why it matters

A “handoff” of information and assets occurs when a work product transitions from one process step to another. And when situations require a specific person or unique input from another person or a team, a dependency occurs. While some handoffs and dependencies are inevitable, excessive and unnecessary dependencies and handoffs, impede flow.

What to do about it

Use the ART planning board to visualize dependencies. The ART planning board helps track significant dependencies across teams and optimize execution throughout the PI. It is adjusted as new facts emerge.

Foster incremental execution of dependencies. A significant handoff from one team to another is fraught with uncertainty, rework, and delays. Instead, dependencies can often be divided into smaller, more manageable items. This approach fosters frequent cross-team integration, proactive discovery of issues and inconsistencies, and a more predictable and faster process.

Synchronize frequently. The ART Sync provides an excellent venue to synchronize dependencies and handoffs. Additionally, establishing direct communication across teams that have dependencies helps sort out implementation details.

Optimize the structure of teams. Excessive dependencies and handoffs often suggest a poorly structured ART. To reduce dependencies, build cross-functional,

cross-disciplinary [Agile Teams](#) and apply team topologies.

Visualize and manage dependencies with external parties. An ART may have internal *and* external handoffs with other ARTs or organizations that should also be identified, tracked, and handled.

#4 Get Faster Feedback

Why it matters

Solution development relies on the fast feedback that guides the ART in the right direction. Mistakes pile up quickly when feedback is delayed or missing, leading to substantial rework for multiple teams, slow delivery, and unsatisfied customers.

Two types of feedback are typically required, as Figure 1 illustrates:

1. Is the ART creating the right thing for the Customer?
2. Is the ART building it right

Figure 1. Building the right thing and building it right.

Figure 1. Building the right thing and building it right

What to do about it

Assure both types of feedback. Continuous customer engagement is the only way to ensure that ART is building the right thing. In many situations, shortcuts to reaching the Customer may be necessary. In addition, the ART must confirm that the technology is viable and that the solution meets quality expectations. This testing should be continuous and address functional and [Nonfunctional Requirements](#) (NFRs).

Provide solution telemetry. Application telemetry can capture valuable system usage and behavior data. These application-specific and context-dependent measures require the planning and execution of enablers.

Engage with customers early and often. At every opportunity, demonstrate the product to the Customer. Test new trial versions of functionality to select customers whenever possible.

Integrate and test frequently. Integration is required within and across the train's Agile Teams. Supported by test automation, it provides the fastest and most productive way to ensure that the implementation is progressing in the right direction.

Use research spikes and MVPs. [Spikes](#) and Minimum Viable Products are examples of intentional, experimental design. They are valuable shortcuts to acquiring knowledge about the Customer and the solution.

5 Work in Smaller Batches

Why it matters

Operating in large batches of work leads to information decay, delayed feedback, rework, and high variability.

What to do about it

Understand the types and respective sizes of batches. There are multiple types in play — planning, integration, testing, release, customer feedback, etc. Each must be understood and sized to minimize the total sum of the transaction and holding cost.

Use cadence. ART cadence naturally constrains a few major batch types. Sticking to recommended iteration and PI duration helps planning, integration, and customer feedback and keeps batch size small.

Manage team and ART size. Applying the recommended size of Agile Teams and ARTs also reduces batch sizes.

Automate a delivery pipeline. An effective Continuous Delivery Pipeline (CDP) reduces optimum integration, test, and deployment batch size.

Plan for smaller batches. Explicitly planning for smaller batches can contribute to reducing their size. For example, planning for specific, frequent releases may help keep the release batch in check.

Use thin vertical slices of work. Thin vertical slices of work can reduce most batch sizes. So do smaller and more manageable features. Taking fewer features within each iteration also reduces batch sizes.

#6 Reduce Queue Lengths

Why it matters

The ART backlog queue contains all the committed feature work awaiting service. The longer that queue, the longer the wait for the new features to reach the Customer.

What to do about it

Keep roadmaps flexible. A fixed long-term roadmap is an example of a long queue. While some milestones must be fixed, the roadmap should keep dates and the scope of work flexible whenever possible. That enables the ART to respond to market changes and new learnings during the implementation.

Establish a strong Product Management function. Often, queues occur because the organization cannot say no and cannot prioritize work effectively. Product managers must exert positive yet firm scope management leadership, demonstrated during PI Planning, the PI itself, and Inspect & Adapt.

Leave capacity for emergent priorities. An ART can plan only for the work it knows about *ahead of planning*. During execution, some areas will require further exploration or depend on future market events. Other events are unforeseeable. As a precaution, allocate reserve capacity so the teams can incorporate it into the plan when an event triggers new work.

#7 Optimize Time ÐIn the ZoneÐ

Why it matters

Solution development relies on creativity and the focused intellectual effort of team members. For example, implementing a new software feature may require developers and teams to navigate hundreds of dependencies between different systems. Cooperation requires a high degree of focus and many work sessions. Optimizing the time spent in the zone — both for individuals and the team together — makes a substantial difference in ART productivity.

What to do about it

Keep work-in-process low. Too much work-in-process (WIP) often results in context switching. Fewer active items at any given time mean fewer work interruptions.

Frequently integrate work. Performing cross-team integration regularly resolves

inconsistencies and issues quickly. Otherwise, they pile up and cause teams with dependencies to interrupt one another frequently and then have to revisit earlier work.

Maintain solution health. If teams don't continuously address technical debt, they will spend too much time chasing ambiguous dependencies and newly-created defects.

Ensure efficient events. Continually optimize all events for team productivity—especially PI planning, systems demos, and I&A.

#8 Remediate Legacy Policies and Practices

Why it matters

As described in the [Team Flow](#) article, legacy policies and practices present pernicious problems that may occur during and even after a SAFe implementation. And it's not just the teams that are affected. Given that ARTs deliver the primary economic value of solutions and consume most of the R&D investment, they are sure to get much attention from critical stakeholders. And unfortunately, some of those stakeholders may not have participated in the Lean-Agile transformation, so they may unknowingly contribute to a lack of flow.

What to watch for

The first step is to know that these impediments are likely to exist. The second is to recognize them when they occur. Common impediments include:

- Traditional project and program management (EVM, WBS, IMS, cost accounting) layered on top of Agile teams
- Quality systems and governance models containing embedded waterfall stage-gate milestones
- Committed scope and hard deadlines that did not include input from the people who will do the work
- Redundant reporting and release meetings with overlapping agendas and attendees
- Separating product design, architecture, and UX from the ARTs and value streams
- Freezing assets to prevent needed changes that customers need
- Slow and ineffective provisioning of required tooling and work environments
- Verification & Validation and compliance activities and teams engaged only at the end

What to do about it

Once identified, teams must take corrective action. But the action depends on the impediment's specific stakeholders, nature, and context. Change agents often discover that many of these stakeholders have not been on the transformation journey and require additional training and coaching. No change journey is

complete until the changes are fully embedded in the culture.

Measuring ART Flow

Measuring flow is critical to improving ART flow. SAFe's [Measure and Grow](#) system provides three measurement categories—*competency*, *flow*, and *outcomes*—that can assess and improve an enterprise's ability to deliver innovative business solutions quickly. This measurement system includes six measures specific to flow: *flow distribution*, *velocity*, *time*, *load*, *efficiency*, and *predictability*. Regarding ART flow, they are all relevant, but predictability, time, and load are particularly useful and highlighted below.

ART Flow Predictability

Art Flow predictability measures how well a train can plan and meet its PI objectives. For a business to plan and execute effectively, ARTs should generally satisfy most of the committed objectives and one or more of the uncommitted. This approach typically results in an average of 80-100% of the total planned. Figure 2 shows an example where the ART is taking a fairly conservative approach to its PI commitments, as the average predictability regularly runs above 100%. Scoring above 100% requires teams to complete their committed and uncommitted objectives.

Figure 2. An example of ART flow predictability

Figure 2. An example of ART flow predictability

ART Flow Time

ART flow time measures the total elapsed time to deliver new features. It's typically calculated from ideation to production. Figure 3 illustrates an example with a reasonably steady flow time average of around 36 days. Outliers are common, as they may indicate features blocked due to outside dependencies, unanticipated risks, or other common factors.

Figure 3. An example of one ART's flow time.

Figure 3. An example of one ART's flow time

ART Flow Load

ART flow load measures the total amount of work in the system at any point. It is often derived as an automated report based on the ART Kanban states, illustrated with an area chart, as shown in Figure 4.

Figure 4. ART flow load example

Figure 4. ART flow load example

This chart illustrates a dramatic increase in WIP for this ART. Additional insights into the ÔwhyÕ would require knowledge of that ARTÕs context. For example, the ART may indeed be overloaded with WIP, or perhaps the size of the ART has increased dramatically, and the increase in WIP is simply a function of size.

ART Flow Efficiency

Flow efficiency is the sum of the active time divided by the total time it takes to get the item through the system. Flow efficiency can be very low in a system that has not been optimized, indicating that most of the time the backlog item spends in the system is waiting for service from another activity. The starting point to understanding flow efficiency is developing a value stream map and estimating (or gathering the actual times) how long a backlog item spends in various process states.

An example is the value stream map in Figure 5 below, taken from an actual DevOps value stream mapping exercise. It can be observed that this system is

highly inefficient, with a total efficiency of 5%. It can also be observed that almost all the waiting time occurs in front of one specific step. That step is the bottleneck in the process, and working to increase the efficiency in other steps will not produce much economic return. The ART must address the bottleneck to increase efficiency, and when that happens, flow time should also decrease dramatically.

Figure 5. An example of ART flow efficiency derived from a value stream mapping exercise

Flow Metrics Summary

SAFe®s Flow metrics will help highlight ART flow improvement opportunities. However, the numbers alone cannot tell the whole story. Qualitative analysis, along with good judgment, is needed.

Last Updated: 20 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Energy flows where intention goes.

~ Rhonda Byrne,
Australian author,
and TV producer

Portfolio Flow

Portfolio Flow describes a state where Lean Portfolio Management provides a continuous flow of new epics to Solution Trains and ARTs to achieve the portfolio's vision and strategic themes.

Adjust Cookie Settings

The LPM competency aligns strategy and execution by applying Lean and systems

thinking approaches to strategy and investment funding, Agile portfolio operations, and governance. This competency has proven to improve business outcomes for SAFe enterprises building the world's most important solutions. But as with any system, flow can always be improved. Improving the flow of customer value through the portfolio is a key economic driver for the enterprise. That is the subject of this article.

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of eight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Details

SAFe defines a set of eight flow accelerators (see [Principle #6, Make Value Flow without Interruptions](#)) that can address, optimize, and debug issues with achieving continuous value flow. Portfolio flow describes accelerating the flow of the significant initiatives ([Epics](#)) needed to accomplish the [Portfolio Vision](#) and advance the [Enterprise](#) strategy.

#1 Visualize and Limit WIP

Why it matters

Systemic overload caused by too much portfolio WIP undermines performance and value delivery. It overloads the ARTs, inhibits responding to change, and results in costly productivity and quality problems, delayed return on investment, employee burnout, and reduced engagement. There is simply no benefit to putting more work into implementation than a portfolio can handle. And nowhere is excess WIP

more damaging than deciding which of the most significant strategic initiatives are needed to deliver effective portfolio outcomes.

What to do about it

Make all significant Epics visible. Ensure that any projects not transitioned to SAFe and all SAFe epics (above the epic threshold) are tracked in the portfolio Kanban. Otherwise, the total portfolio WIP may be underestimated.

Review and adjust the epic threshold to exclude non-epics. Epics are sometimes misused as “important things we want to discuss forever.” Ensure that only significant initiatives above the epic threshold guardrail enter the portfolio Kanban and that there are some reasonable time-bound expectations. Adjust the threshold when needed, which may require including additional criteria.

Review and validate portfolio Kanban WIP limits. If there are too many epics in progress, adjust WIP limits. Consider adding classes of service to enable high-priority items to go quickly through the system while lowering the overall WIP.

Understand the capacity of each value stream and ART. Collect metrics for each ART’s capacity to prevent more work from going to the teams than they can reasonably analyze and implement.

Ignore sunk costs. Stop significant initiatives if they no longer support the strategy or are not providing the intended value.

#2 Address Bottlenecks

Why it matters

Portfolio gridlock can create a bottleneck that hinders the review, analysis, approval, and implementation of the most significant initiatives needed to achieve the portfolio’s vision. The resulting delays threaten the downstream performance of all the [Development Value Streams](#). That creates a choke point that limits portfolio value delivery and imperils strategy.

What to do about it

Help ensure LPM has the proper decision-making authority. Verify that the LPM team has sufficient time and authority to participate and make portfolio decisions without escalation.

Increase the pool of Epic Owners. A common cause of portfolio bottlenecks is a lack of sufficient [Epic Owners](#). Leverage the knowledge of subject matter experts, [Business Owners](#), [Product Managers](#), [Product Owners](#), and [Enterprise Architects](#) to serve as Epic Owners.

Understand ART capacity. Portfolio work can’t go anywhere unless there are people to do it. Understand the capacity of each affected ART to take on this new work in a reasonable time frame.

Ensure the Lean Business Case is lean. Despite moving to SAFe, many businesses still tend to create traditional business cases that can take too long to develop. Their length and detail, speculative financials, and unproven assumptions may cause questions and delays during analysis.

#3 Minimize Handoffs and Dependencies

Why it matters

Portfolio workflow is critical, but the mechanism of the portfolio kanban is not particularly complex. Typically, there are only a few dozen epics in flight, and they usually require a limited number of stakeholders with the decision authority to advance them. However, managing the work through the portfolio Kanban often requires different skills and people. Some degree of handoffs and dependencies will occur. Bringing the right people into the discussion at the right time facilitates portfolio flow.

What to do about it

Properly support the Epic Owner. Epics require review, collaboration, and the engagement of multiple stakeholders who are often busy with their functional roles. Help ensure Epic Owners can reach the people they need promptly.

Understand when cross-value stream coordination is required. Some epics will touch multiple value streams. This coordination increases dependencies and handoffs even during the early stages of epic review and analysis. Leverage Epic Owners, [Release Train Engineers](#) (RTEs), and Product Management to help decide how this new work can be managed across value streams.

Recognize the need to refactor value streams. Much of the new and innovative work that realizes the evolving strategy will arise in the portfolio backlog. During analysis, LPM, with the help of the VMO or [LACE](#), can advise on what value streams may need to be created, adjusted, or eliminated. That, in turn, reduces handoffs and dependencies in the new work.

#4 Get Faster Feedback

Why it matters

Faster [Customer](#) feedback, especially in the early stages, is vital to evaluate new initiatives quickly. This feedback ensures epic investments do not result in designing and building solutions that Customers do not want or require a business model change that the enterprise cannot achieve.

What to do about it

If epics require changes to the business model, test the assumptions. Epics that impact the business model often come with significant opportunities and big challenges. During epic review and analysis, test business model assumptions with business owners, executives, and Customers helping avoid unrealistic assumptions, false starts, and future sunk costs.

Validate viability by engaging with customers early. While solution development resides with the teams, LPM stakeholders need fast feedback about these new initiatives to support decision-making. These decisions should happen long before the MVP by testing early mockups and low-fidelity prototypes. Additional Customer insights should also be captured in the Lean business case.

Focus on the leading indicators for MVPs. Verify that epics employ the right leading indicators to measure their progress toward the business outcome hypothesis.

#5 Work in Smaller Batches

Why it matters

Smaller batches go through any system more quickly and with less variability, fostering faster learning. These smaller batches are especially important for the portfolio because epics are significant initiatives that can take a long time to review, analyze and approve. If the batch size is too big, portfolio stakeholders may not have time to evaluate all the work in flight responsibly. This creates delays in decision-making and clogs the flow of value through the portfolio Kanban system.

What to do about it

Limit the number of epics reviewed during LPM events. Experiment with how many epics LPM can realistically be considered during the LPM events, such as the Portfolio Sync. Then, adjust the limit based on context and history.

Reduce the transaction cost of reviewing and analyzing epics. For example, timebox each epic review to ensure discussions remain clear and concise. This also has the beneficial effect of increasing decentralization and fostering trust.

Conduct low-fidelity tests during analysis. Perform research spikes (explainer videos, landing pages, interviews, paper prototypes, fake product sign-ups, micro-surveys, etc.) to understand better the Customer's problems and the range of solutions.

Leverage a common cadence. Applying the same PI cadence across all value streams in the portfolio can help reduce batch sizes, providing a regular rhythm for reviewing portfolio work.

Reduce experiment size. Wherever possible, reduce the effort and scope of work needed to get earlier feedback.

#6 Reduce Queue Lengths

Why it matters

Long queues of portfolio work reduce strategy responsiveness and can cause the enterprise to miss critical market windows. This can have a direct effect on the overall competitiveness of the enterprise.

What to do about it

Reroute non-portfolio work immediately elsewhere. The portfolio can be a ÒcatchallÓ for Òeverything important that needs to be discussed.Ó Challenge why each item needs to be in the portfolio. Limit the portfolio queue to true epics that require portfolio attention due to their investment or impact.

Understand and attend to critical market events and market rhythms.

Buying is an act of the Customer, not the producer. Portfolio epics must always be considered in light of the time criticality of the initiative from the CustomerÕs perspective.

Eliminate non-strategic and bad ideas quickly. Since more effort and capacity are needed as an epic moves through the Kanban, LPM should quickly decide which will proceed or be removed.

Replace fixed schedules with flexible roadmaps. Inflexible roadmaps create long queues and delays in introducing new critical work. Limit longer-range commitments and replace fixed plans with flexible rolling-wave roadmaps.

#7 Optimize Time in ÒThe ZoneÓ

Why it matters

It can be challenging for busy executives to find sufficient time to develop and evolve strategic plans *and* engage in the ongoing work of LPM. It requires significant intellectual and mental energy, as free as possible from daily business distractions. In other words, itÕs hard to find the Òtime in zoneÓ necessary to collaborate with peers who share responsibility for strategy. What results is a flawed strategy, communicated ineffectively or established and rolled out intermittently versus a continuous evolution.

What to do about it

Leaders, change agents, and coaches need to facilitate a process and experience that creates time in the zone to formulate and execute strategy. Although itÕs one of the primary purposes of LPM and its various events, those meetings alone may not provide enough time. Additional focus is required.

Ensure sufficient time is allocated to develop the strategy. Consider holding

separate portfolio strategy workshops to focus solely on reviewing the portfolio strategy, vision, [Lean Budget Guardrails](#), and business outcome metrics.

Hold effective Portfolio events. Ensure the Strategic Portfolio Review, Portfolio Sync, and Participatory Budgeting (see [LPM](#)) events are effective and held regularly. Ensure that the data presented are current and provide insights to facilitate decision-making by the right individuals.

Recognize when a Portfolio Epic no longer needs LPM focus. LPM and portfolio stakeholders have limited time. Ensure the focus is on epics that require LPM oversight and focused attention. Move epics to “done” when they are rejected or as soon as they are no longer a portfolio concern.

Eliminate redundant portfolio governance practices. Epics receive increased scrutiny because of their size, cost, and impact. Replace traditional reporting with SAFe’s Lean-Agile practices that measure progress objectively, including leading indicators and [KPIs](#), to ensure efficient LPM focus.

Invest in meeting facilitation to optimize the “Ozone.” Effective facilitation will address timeboxing, personalities, and misalignment. It can also create a “must-attend event” where people say no to any potential cancellation or lack of attendance. Effective facilitation can also help eliminate or address emotional drama or misalignment amongst key stakeholders.

#8 Remediate Legacy Policies and Practices

Why it matters

Moving to LPM is a significant change for any enterprise. And during the transition, many companies will likely manage some current initiatives by applying their existing traditional governance policies. This means that two different oversight methods will attempt to coexist, resulting in people reverting to the old process for even the new initiatives. This creates an ongoing burden for the enterprise, increases overhead, and slows the flow of value.

What to watch out for

The remedy is to recognize that many historical impediments probably exist and to address them when they occur. Common obstacles include:

The portfolio Kanban becomes a central intake for all work, not just new epics that exceed the Lean budget guardrails

Funding projects instead of value streams

Continuing to forecast and report capital and operating expenses based on outdated practices and manual timesheets

Quarterly strategic planning is disconnected from [PI Planning](#) and the LPM portfolio review process and includes different stakeholders

Making significant customer commitments without engaging with the teams who will

do the work or understanding the impact on in-flight or committed work
Overly-detailed business cases that require too much upfront investment, burdening and limiting solution design
Limiting access or visibility to the portfolio Kanban, causing misalignment, uncertainty about downstream priorities, and redundant work
Running traditional program management in parallel or on top of Agile practices. (For example, waterfall phase gates, change control boards, project cost accounting, and outdated status reporting)
Incongruent and conflicting policies in finance, HR, etc.
Mindsets that defend sunk costs or ongoing commitment to legacy products and services that prevent pivoting to new opportunities

What to do about it

Identify the legacy portfolio activities that should be stopped or replaced. Take appropriate action based on root cause analysis. Ensure reasonable fidelity to the SAFe LPM process.

Measure and Improve flow

SAFe's [Measure and Grow](#) guidance offers portfolios a way to assess and improve their ability to deliver innovative business solutions quickly. It includes six measures specific to flow: *distribution, velocity, time, load, efficiency, and predictability*. Flow *time, load*, and *distribution* are particularly relevant to the portfolio and are described below.

Flow Time

Flow time measures the interval needed for all the steps in a defined workflow to be completed. Portfolio flow time can be estimated from ideation to production. Or for example it can also be helpful to measure portfolio flow time from epic is pulled into the "reviewing" state until its hypothesis has been evaluated.

Figure 1 shows an example of portfolio epic flow time. In this example, flow time represents the period it took an epic to go from "reviewing" to "done". (Note, "done" doesn't mean the implementation is complete, but it is no longer a portfolio concern.) The scatter plot illustrates that 99% of epics had a flow time of fewer than 261 days, 75% fewer than 68 days, and 50% had a flow time of fewer than 115 days.

Figure 1. An example of portfolio flow time

Figure 1. An example of portfolio flow time

Flow Load

Figure 2. Example Flow load for a portfolio illustrated in a CFD

Figure 2. Example Flow load for a portfolio illustrated in a CFD

Flow load indicates how many items are currently in the system. Keeping a healthy, limited number of active items in portfolio WIP is critical to enabling the fast flow of strategic value. Figure 2 illustrates a Cumulative Flow Diagram (CFD) that depicts the flow load of epics for a specific time frame. The vertical line highlights the current low load, which excludes the “done” state.

Flow Distribution

Flow distribution measures the amount of each type of work in the system at a specific time. A helpful view of portfolio flow distribution illustrates the trend of money allocation across investment horizons.

Figure 3. An example value stream's flow distribution for investment horizons over time

SAFe's flow metrics can highlight improvement opportunities for portfolio work. However, these measurements cannot tell the whole story alone. Qualitative analysis is required to provide context for the flow metrics and further understand the portfolio's current state.

Learn More

[1] <https://www.inc.com/jessica-stillman/3-tricks-to-help-you-get-in-the-zone.html>

Last updated: 25 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)

- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Whenever there is a product for the customer, there is a value stream. The challenge lies in seeing it.

~Learning to See [1]

Operational Value Streams

[Adjust Cookie Settings](#)

An Operational Value Stream (OVS) is the sequence of activities needed to deliver a product or service to a customer.

Examples include manufacturing a product, fulfilling an order, admitting and treating a medical patient, providing a loan, or delivering a professional service. Most employees in the enterprise work in operational value streams that serve the ultimate customer directly, where they may:

- Market the enterprise's products and services
- Sell and process orders
- Manufacture products
- Provide customer support and deliver related services

The people that *develop* the products and services work in [Development Value Streams](#), the subject of the companion article, Development Value Streams. Understanding the enterprise's OVS is integral to effective solution design and implementation. These people use the solutions; knowing them means knowing your customer, a critical aspect of [Customer Centricity](#) and [Design Thinking](#). OVS offers products or services to Customers, which helps keep the business profitable and healthy.

Details

Value streams are the most fundamental construct of Lean thinking and are foundational to SAFe. Lean thinking can be summarized as follows: [2]

- Precisely specify value by product
- Identify the value stream for each product
- Make value flow without interruptions
- Let the customer pull value from the producer
- Pursue perfection

There are two types of value streams in SAFe, *Operational* and *Development*.

Each value stream represents the sequence of steps an enterprise uses to deliver value to its customer. Each highlights the flow of value, delays, rework, and bottlenecks. They illustrate how the current flow affects the people who do all the work. Identifying, visualizing, and optimizing value streams is the primary method a Lean enterprise employs to shorten time to market while improving the timeliness, quality, and value of its products and services.

Defining Operational Value Streams

Whether obvious or not, every product or service already has an OVS, the series of

steps used to deliver a specific product or service to the customer. For example, fulfilling an order, admitting and treating a medical patient, or providing a loan or professional service.

Figure 1 illustrates the structure and flow of a value stream. A "trigger" usually a request for a product or service, initiates the flow. Each chevron identifies a "step" an activity needed to process the order. Each step takes time to complete. The sum of all the processing-step times, plus the delay periods between them, is the "total lead time." Shortening the lead time is the fastest way to reduce the time to market.

Figure 1. Example and structure and flow of an OVS

Figure 1. Example and structure and flow of an OVS

Value streams are persistent and enduring for as long as customers continue to place orders for their products or services. They cut across departments and functions, and each contains:

1. All the *steps* necessary to convert the trigger to the delivery of value
2. The *people* who perform these steps

3. The *systems* they use to do their work
4. The flow of *information* and *materials* that are necessary to satisfy that request

Comprehending the four steps above is critical to understand how value flows to customers. Figure 2 illustrates this expanded view of an OVS.

Figure 2: An expanded view of an operational value stream

Figure 2: An expanded view of an operational value stream

Types of Operational Value Streams

There are four common *OVS* patterns:

1. **Fulfillment** represents the process of processing a customer order for a digitally-enabled product or service by delivering it and receiving payment. Examples include providing a consumer with an insurance product or fulfilling an eCommerce sales order.
2. **Manufacturing** converts raw materials into the products customers purchase.

Examples include consumer products, medical devices, and complex cyber-physical systems.

3. **Software products** offer and support software applications and solutions for sale to the end user or enterprise. Examples include ERP systems, SaaS, and desktop and mobile applications.
4. **Supporting** value streams are end-to-end workflows for various repetitive and internal support activities. Examples include employee hiring, establishing and executing supplier contracts, performing the annual audit, and completing an enterprise sales cycle.

Figure 3 illustrates an example of each type of OVS.

Figure 3. Examples of the four types of OVS

Figure 3. Examples of the four types of OVS

And although companies need functional departments to build and share knowledge, they are not value streams since they don't deliver value from trigger (request) to delivery of the product or service to the ultimate customer. However, many of the people in these departments participate in one or more OVS.

It stands to reason that large enterprises typically offer their customers various products and services. That's one of the ways they grow. As such, it follows that there are a substantial number of value streams within those enterprises. Figure 4 illustrates how a *consumer loan* value stream is just one of a large bank's offerings and OVS.

Figure 4. Highlighted operational value stream for consumer loans in a commercial bank

Identifying Operational Value Streams

Value streams provide the most essential and fundamental knowledge of *how* an enterprise serves its customer. There is no substitute, and the Lean enterprise continually improves its business performance by identifying, analyzing, and optimizing its value streams. The purpose of DVS is to create and advance the systems and products for the OVS. So, understanding those value streams is also

indispensable to business performance.

However, unlike the footpath lighting in a dark theater, value streams do not illuminate themselves. They are complex, and perhaps no one in the enterprise understands any single OVS flows exactly. This means that [SAFe Practice Consultants \(SPCs\)](#) and their Lean-Agile leaders often are asked to understand and help optimize the organization's OVS that they serve. The following questions can help with that analysis:

- Who are your users and customers? Are they internal, external, or both?
- What products and services do you market, sell and support?
- What solutions do you provide for internal users?
- What triggers the flow of value?
- What value do your customers perceive you deliver?

Once identified, the template in Figure 5 can be used to capture a particular value stream's purpose and attributes.

Figure 5. Value stream definition template with consumer loan example

Figure 5. Value stream definition template with consumer loan example

Operational Value Streams and Customer Journey Maps

As shown in Figure 6, customer journey maps are powerful design thinking tools. They illustrate the user experience as customers engage with a company's OVS and its products and services. They allow teams to identify ways one or more DVS can improve the customer journey, creating a better end-to-end experience.

Figure 6. Customer journey map for the consumer loan value stream

Figure 6. Customer journey map for the consumer loan value stream

Many developers build and support *internal* systems that OVS uses to create the customer's journey. Improving these underlying systems improves the *external* customer's journey. As such, the people in the OVS who use these internal

systems are DVS's customers. [Customer centricity](#) and [Design Thinking](#) applies there equally well.

Improving Operational Value Streams

Finally, value streams enable ***value stream mapping***, a collaborative process in which a group of stakeholders defines a value stream's steps, handoffs, and delays. This mapping highlights the total lead time needed to fulfill a request while spotlighting areas for improvement.

Figure 7 illustrates a value stream map for a marketing campaign that supports a product launch.

Figure 7. Value stream map for a marketing campaign

Figure 7. Value stream map for a marketing campaign

With most such value streams, the wait ***between*** steps consumes most of the lead

time. Indeed, it is not unusual that the total processing time is as little as 5% of the lead time.

The natural tendency to improve lead time is to reduce *processing* time, which has obvious benefits. After all, that's where the people do the work. And everyone wants to work more efficiently. But excessively focusing on processing time can negatively impact people, culture, and quality. That, in turn, can *increase* overall lead time and decrease economic outcomes.

But no one likes waiting. So, focusing on *reducing the wait time between steps is more effective*. Much of the guidance in SAFe focuses directly on this problem.

[Principle #6 D Make value flow without interruptions](#), provides some of the practical ways to do this.

Learn More

[1] Rother, Mike, and Shook, John. *Learning to See. Value Stream Mapping to Create Value and Eliminate Muda*, 20th-anniversary edition. Lean Enterprise Institute, 1998-2018.

[2] Womack, James, and Daniel Jones. *Lean Thinking; Banish Waste and Create Wealth in your Corporation*. Simon and Schuster, 1996-2003.

Last update: 19 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

[Download SAFe Posters & Graphics](#)

Watch and download
SAFe videos and
presentations

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The aim of development is, in fact, the creation of profitable operational value streams.

—Allen Ward [1]

Development Value Streams

A Development Value Stream is the sequence of activities needed to convert a business hypothesis into a digitally-enabled solution that delivers customer value.

As described in [Principle #10, Organizing around value](#), the value stream concept is a critical underpinning of Lean thinking and is fundamental to SAFe. There are two types of value streams described in SAFe.

Development Value Streams (DVS). This article describes DVS—the sequence of activities to develop and support the solutions used by [Operational Value Streams \(OVS\)](#). One or more [Agile Release Trains](#) constitute each DVS.

Operational Value Streams (OVS). The OVS is the sequence of activities needed to deliver a product or service to a customer. Examples include manufacturing a product, fulfilling an order, admitting and treating a medical patient, providing a loan, or delivering a professional service.

Systems and software developers, product managers, engineers, scientists, and IT practitioners work primarily in the DVS. External customers use the solutions they create, but many are often *internal* to the business. These customers (or users) are the people within the enterprise who use those systems to do their job. Others offer products, services, or systems developed by the DVS to *external* customers.

Details

A SAFe [Portfolio](#) defines and governs a set of DVS. Each DVS is committed to building, deploying, and supporting [Solutions](#) the enterprise needs to accomplish its business mission (Figure 1).

Figure 1. A SAFe portfolio defines and governs a set of DVS

Figure 1. A SAFe portfolio defines and governs a set of DVS

Lean Budgets and Guardrails support them and empower the workers who make the day-to-day decisions that optimize economic value. This value is measured by Key Performance Indicators (KPIs), Objectives and Key Results (OKRs), and SAFe's three measurement domains (*Outcomes*, *Flow*, and *Competency*, see Measure and Grow) which are established to evaluate how the value stream is performing in achieving the portfolio's strategy.

Why Organize People in Development Value Streams?

Simply put, organizing people around the DVS improves workflow, efficiency and accelerates time to market. This is accomplished by optimizing the flow of value to the Customer across divisions and functional departments through suppliers, channels, and the whole system.

Value streams offer many benefits, as they:

- Enable long-lived, stable teams that focus on delivering value
- Help identify and visualize all the work necessary to produce solutions
- Create transparency of delays, bottlenecks, and handoffs
- Support smaller batch sizes of work
- Enable knowledge growth and more continuous learning
- Allow the funding model to shift to value streams from traditional project budgets

Indeed, when you start to understand their worth, it makes you wonder how enterprises ever got along without them. Yes, they were always there, but we didn't **see** them. [2]

Defining Development Value Streams

Value streams contain all the activities, people, systems, and the flow of information and materials needed to deliver value. While OVS vary significantly depending on their purpose, the DVS steps are relatively standard. Figure 2 illustrates the simplified structure of a DVS.

Figure 2. Example structure of a DVS

Figure 2. Example structure of a DVS

This structure contains the following elements:

Trigger ⚡ the value stream is *triggered* by a new request, though many new requests are moving through the value stream at the same time.

Steps ⚡ contain the activities needed to define, build, validate, and release new value.

Bar ⚡ the bar between the steps indicates information and materials flowing from one process to the next. It also implies the typical handoffs of information that occur as people in different steps add value to the process.

Ellipses (·) ⚡ indicate the delays between these steps, typically the most significant contributors to long lead times. Decreasing delays is usually the fastest and most efficient way to reduce lead time.

The output of the DVS is a new *increment of the solution*, the additional value these new features provide.

This example DVS in Figure 2 is an incredibly simplified model of what it takes to

create innovative technical solutions in today's digital enterprise. Still, it serves an important purpose: it illustrates the activities needed to deliver solutions and the time it takes.

The [Operational Value Streams](#) article describes how value streams deliver solutions and support the Customer's needs. Indeed, as Ward notes, [1] the entire purpose of development value streams is to make operational value streams—and, thereby, the whole enterprise—more profitable and more efficient in delivering value. As that article illustrates, four common OVS value stream patterns—fulfillment, manufacturing, software products, and support—lend themselves to different types of DVS patterns.

1. Fulfillment DVS Pattern

The first pattern (Figure 3) is a consumer loan example. Patterns like this are fairly standard in insurance, banking, financial services, and related industries that offer complex, digitally-enabled products and services to consumers (B2C) and businesses (B2B).

A fulfillment OVS represents the steps to process a customer request, deliver a digitally-enabled product or service, and receive payment. Examples include providing a consumer with an insurance product or fulfilling an eCommerce sales order (Figure 3).

Figure 3. Fulfillment DVS pattern for a consumer loan

Figure 3. Fulfillment DVS pattern for a consumer loan

In this case, the product is more virtual than tangible, as the “loan product” is a set of commitments, interfaces, applications, services, contracts, licenses, and other relationships that constitute the consumer product or service.

The Customer interacts at various points throughout their journey. Although critical, customer access points are only the tip of the development iceberg; most of the development happens on internal enterprise-class systems, like those depicted for the commercial banking system.

To build and maintain these systems, multiple development streams may be required. For example, one development value stream supports the front-end loan origination services and credit scoring; another builds the core banking services.

2. Manufacturing DVS Pattern

Figure 4 illustrates a pattern for manufacturing a significant cyber-physical system, in this case, a passenger vehicle. This pattern shows fundamental differences between the development streams that directly support digital solutions and those that support products that must be manufactured before use.

In this case, the delivered value is not the product, but the specifications needed to manufacture and validate it. The focus is on **Solution Intent**, the repository of design specifications, manufacturing procedures, bills of materials, and so on, required to produce the device.

In this context, teams serve two types of customers:

1. The ultimate Customer is the end-user of the manufactured product (the driver of the vehicle in this case)
2. The manufacturing personnel who use the specifications to build the product (illustrated with the blue Œshop foremanŒ icon).

Figure 4. Example DVS pattern for manufacturing

Figure 4. Example DVS pattern for manufacturing

Of course, the size and the number of DVS vary with the solution's complexity. In some instances, thousands of people are involved in designing such a system (vehicle, aircraft, satellite, smartphone, and so on). But it's also the case that many manufactured products can be developed by a single development value stream (drone, webcam, remote control, and so on).

The more complex case pictured above illustrates another common sub-pattern in which some DVS directly support others. In this example, one DVS is devoted to building vehicle designers' tools needed by another DVS to design, simulate, and validate their products.

Also pictured is a "digital twin," a replica of the product used to validate design assumptions. This twin is a common Lean-Agile strategy for building significant cyber-physical systems.

Software Products DVS Pattern

Many software development and IT practitioners work in the Independent Software Vendor (ISV) industry, where vendors directly produce and market software products. The software products segment includes the largest digital-native companies and hundreds of thousands of enterprises that sell everything from IT services to hospital information systems, desktop software, gaming, and simple mobile apps.

Figure 5 illustrates a pattern for a significant enterprise that develops and supports a substantial software application.

Figure 5. Software products value stream

Figure 5. Software products value stream

This illustration shows how the largest focus of development effort goes directly into the software solution. The customers (and various customer personas) are users of the system. Hundreds or even thousands of people may be directly involved in developing, deploying, and maintaining such systems.

But the system doesn't sell itself, answer support calls, or collect revenue. The OVS is responsible for customer acquisition, internal operations, support, and more. As pictured, some software and IT practitioners support and maintain those internal systems. It's also important to note that customers interact with the company throughout their buyer journey while using the application. As illustrated here, a DVS is devoted to supporting that functionality. It could be distributed to the other development Agile Release Trains (ARTs) for a more stream-aligned end-to-end approach. However, a common approach to the customer experience may warrant a separate development value stream.

4. Supporting DVS Pattern

The final type of OVS serves Ôsupporting value streams,Õ the internal and critical functions, and the people and processes that keep the enterprise running. Common examples include the annual audit process, hiring and onboarding personnel, and many other significant, repeating workflows. Additionally, enterprises engaged in logistics, supply chain, research, data mining, drug discovery, and so on have extensive and critical internal supporting value streams, with activities far earlier than those aimed at the end customer. Many such supporting value streams exist, creating substantial demand for development.

Often, multiple OVS can be supported by one DVS that builds, configures, and supports the systems the OVS needs to function and share standard information.

Figure 6 illustrates an example of a single DVS that supports an ERP system used throughout the enterprise.

Figure 6. Example of a DVS that supports multiple supporting OVS

Figure 6. Example of a DVS that supports multiple supporting OVS

In this example, there is only an internal Customer. That's because, in this case, the DVS supports an OVS internal to the enterprise. The Customers for the DVS are the users, stakeholders, and employees who work in the OVS (indicated by the circled people icon).

Understanding this internal flow of value is as critical as understanding the external Customer. The mindset, methods, and practices of [Customer Centricity](#) and [Design Thinking](#) apply equally to the teams on this DVS.

Defining Development Value Streams

The four OVS patterns described earlier help define the DVS structure for optimal value delivery. Once initially identified, additional analysis is required to determine the DVS boundaries, people, solutions, and other deliverables. Figure 7 provides a *Development Value Stream Canvas*, a simple template that can capture and refine the emerging understanding.

Figure 7. Development value stream canvas [3]

Figure 7. Development value stream canvas [3]

Optimizing Development Value Streams

Finally, there is another significant benefit to value stream analysis. Each value stream provides a customer with an identifiable and measurable flow of value. As such, “value stream mapping” [2,4] can be applied to measure and improve delivery velocity and quality systematically.

As teams and ARTs begin adopting [DevOps](#), the learning from the journey is often surprising. Figure 8 illustrates an example of a team doing their first mapping of a typical feature as it moves from definition to production.

Figure 8. Results from one team's development stream mapping exercise

Figure 8. Results from one team's development stream mapping exercise

The flow metrics for the value stream offer strong evidence that substantial improvement is required. *Only 5% of the lead time was value-added activity time. The other 95% was spent waiting!* Also, there was significant rework; just **36%** of the features made it through the DVS without it being returned to an upstream process.

These results provide ARTs with a good business case to invest in value stream mapping and further automating the [Continuous Delivery Pipeline](#), as shown in Figure 9.

Figure 9. Development value stream with a Continuous Delivery Pipeline

Figure 9. Development value stream with a Continuous Delivery Pipeline

The continuous improvement process enables the DVS to predictably deliver innovation and value in the shortest sustainable lead time with the highest possible quality.

Learn More

[1] Ward, Allen. *Lean Product and Process Development*. Lean Enterprise Institute, 2014.

[2] Rother, Mike, and John Shook. *Learning to See: Value Stream Mapping to Create Value and Eliminate Muda*, 20th-anniversary edition. 1998-2018. Lean Enterprise Institute.

[3] Thanks to SAFe Fellow Mark Richards for contributing to the Value Stream Canvas concept.

[4] Martin, Karen, and Mike Osterling. *Value Stream Mapping*. M Graw Hill, 2014.

Last update: 14 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*Click. Boom.
Amazing!*

Steve Jobs

Solution

A Solution is a product, system, or service that provides value to internal or external customers.

All the words, pages, roles, activities, and artifacts in SAFe exist for one purpose and one purpose only: to help Agile teams continuously deliver solutions that provide value to the *Customer* and the *Enterprise*. In turn, that enables customers to

Adjust Cookie Settings

achieve their goals.

However, value isn't guaranteed even when teams and trains apply SAFe guidance. After all, customers do not buy [Capabilities](#) or [Features](#); they buy *whole-product* solutions that deliver their desired outcomes. Understanding a solution becomes essential to understanding value delivery in SAFe.

Details

In SAFe, solutions deliver the [Portfolio](#)'s value. Therefore, accelerating solution delivery is vital for organizations to survive and thrive in the [Digital Age](#). Instead of focusing on projects and defining success as completing them *on time and within budget*, enterprise leaders must emphasize *solutions that deliver business outcomes*. This change requires the new organizational and management structures that SAFe advocates, moving from Projects to Products. [1]

In SAFe, the term "solution" is intentionally general, defined as *a product, system, or service that provides value to internal or external customers*. A solution can be a small mobile application built by a single [Agile Release Train \(ART\)](#) or a large automotive system of systems built by a network of [Development Value Streams \(DVSs\)](#) in a supply chain. It may also be an insurance or banking product offered by a financial institution. Solutions can be the products a company sells or the internal products they use to run the business. They may provide direct value to an end-user or may be a component of a larger solution.

Informed by the solution's [Vision](#), [Backlog](#), and [Roadmap](#), practitioners in a DVS define, build, validate, and release solutions to customers, as shown in Figure 1.

Figure 1. A SAFe solution in its context

Figure 1. A SAFe solution in its context

Figure 1 also highlights a solution's four fundamental properties. Solutions are:

Desirable ▷ Do customers and users want the solution?

Feasible ▷ Can we deliver the right solution through a combination of build, buy, partner, or acquire endeavors?

Viable ▷ Is the way we build and offer the solution creating more value than cost?

Sustainable ▷ Are we proactively managing our solution to account for its expected product-market lifecycle?

The remainder of this article describes these four solution properties.

Solutions are Desirable

Solutions are desirable to the portfolio stakeholder who invest in them, the customers who purchase them, and the end-users who interact with them, as described below.

Solutions Deliver the Portfolio's Value

Enterprises build many solutions and have infinite opportunities to build others. However, just because an organization *can* build a solution does not mean it *should*. Portfolio leaders are responsible for determining which solutions best support the portfolio's strategy.

The [Portfolio Vision](#) defines the DVSs that lead to customer solutions and shows how their performance ([KPIs](#)) will deliver the portfolio's objectives defined by the [Strategic Themes](#) OKRs (Figure 2).

Figure 2. The Portfolio Canvas connects strategy to execution and shows how solutions will deliver the portfolio's value and realize its objectives

In SAFe, all DVSs build solutions with committed teams and funding (see [Lean Budgets](#)) that remain so long as the solution is in operation.

Solutions Deliver Value to Customers

Solutions in SAFe solve a business need, delivering value to an internal or external customer. Some solutions are products the enterprise sells directly to customers. Others may optimize the organization's internal operations. But every solution has one or more customers who recognize its value.

To build desirable products that connect with customers and address their needs, solution builders apply a [Customer-Centric](#) mindset. Some customers are direct, end-users of the solution. Others are indirect customers who pay for and specify the solution but are not users. To understand and empathize with all customers, Solution builders apply [Design Thinking](#) to ensure the solution is desirable to all.

Solutions May Support Operational Value Streams

Rapid technology innovations continue to disrupt operational environments. Machines are quickly automating many steps in the [Operational Value Stream \(OVS\)](#), performing them faster and more reliably while reducing costs.

Some solutions directly support an organization's OVS. For example, a banking operation may provide a mobile application for external customers and call center applications for internal customers. Other solutions are products for a customer's operational context. For example, an airline uses an aircraft, and many other solutions, in their passenger-delivery operations.

Solutions are Feasible

Most organizations build and deliver products in their domains. Automotive companies provide transportation solutions, and banking organizations offer financial solutions. However, digital disruption allowed businesses to enter and disrupt new markets. For example, Apple outsold the entire Swiss watch industry just four years after launching the Apple Watch. To compete in the digital age, organizations must have the technical knowledge to build innovative solutions and evolve them to meet changing customer and market demands rapidly.

Solution Building Requires Technical Expertise

Building innovative solutions require a diverse range of skills and technical resources. Organizations need the right technical teams, strategic relationships with the right partners, and economic models that ensure that solutions remain feasible over their lifecycles.

To leverage values and practices that continually encourage individuals to increase

their knowledge and competence, Lean-Agile enterprises adopt a [Continuous Learning Culture](#). Organizations may partner with suppliers, cultivate competence internally, or both, to fill in gaps. Partnering with suppliers brings domain knowledge, practical experience, and pre-existing solution components to the teams. Note: Organizations are less likely to partner when the supply chain innovates too slowly to support the business strategy or the technology is strategically important. In these cases, competence is developed internally through training, hiring, and strategic acquisition.

Modular Solutions Accelerate Value Delivery

Modular designs that communicate through defined interfaces allow ARTs and teams to evolve their parts of the solution independently and accelerate value delivery. Some solutions provide direct value to an end-user; others are modules (also called components or subsystems) that are parts of a larger solution. Solutions are often built from other solutions to help accelerate value delivery, reduce development costs, and improve quality. Those other solutions come from various sources, including internal and external suppliers and open-source communities (Figure 3).

Figure 3. Solutions may provide direct value to a user or may be part of a larger solution

Architects intentionally decompose solutions into modules that teams and ARTs can independently design and deliver. Modules are:

Decomposed: Solutions are split into independent parts that communicate through managed interfaces. Applying a [Domain-Driven Design](#) produces solutions that are easier to change, test, and incrementally develop.

Nested: Modules may contain other nested modules, resulting in a hierarchy. Layered systems are a common pattern for nesting modules.

Integrated: Nested modules are combined for deployment and execution to support all levels of testing and operation. The [Continuous Delivery Pipeline \(CDP\)](#) enables this process.

Distributed: Modules may execute on multiple devices or locations to reduce response times, provide higher throughput, provide fault tolerance, and support other [Non-Functional Requirements \(NFRs\)](#).

Some modules are independently releasable elements called *value streamlets* (see

[Release on Demand](#)) that teams can independently evolve, deploy, and release without waiting for other teams.

Figure 4. Value streamlets can be independently deployed and released for faster delivery and feedback

Solution Intent and Context Define and Evolve the Solution

Solutions have intent and context. The [Solution's Intent](#) defines critical requirements and design constraints, including key decisions. The [Solution Context](#) describes aspects of the operational environment for solution installation, usage, and support. Together, intent and context guide the solution's implementation.

Solution builders apply [Lean Systems Engineering](#) and SAFe [Principle #3 - Assume variability; preserve options](#) by specifying the solution's intent and context incrementally. Some decisions, known early in the solution's lifecycle, are fixed. Others may vary as teams gain knowledge by exploring alternatives using [Set-Based Design](#) to find optimal implementations.

Solution intent and context influence the solution's backlogs and roadmaps in two ways. Fixed decisions drive the work and the backlog items that implement them. Uncertainty also requires work to explore alternatives that can move decisions from variable to fixed. Backlogs contain both types of work to simultaneously build the known parts of the solution while exploring its unknown parts.

Solutions are Viable

The solution's value to the enterprise must offset the costs of building and operating it (Figure 5). Typically, costs are simple to quantify, including development costs, operating costs, and licensing fees, to name a few. Quantifying value, however, can be trickier. Solutions can provide multiple types of value to the organization:

- Monetary
- Improved operations
- Market share maintenance or expansion
- New data and insights about consumers and the operational environment

Figure 5. A proper value exchange model ensures viable solutions

Figure 5. A proper value exchange model ensures viable solutions

Whole Product Thinking Ensures Long-Term Viability

To realize the solution's value, customers need more than great features. They also need excellent service, pre-sales support, documentation, training, reasonable pricing, a promising roadmap from a reputable company, and more. Customers expect these to be part of the overall offering.

As shown in Figure 6, Whole-Product Thinking (see [Customer-Centricity](#)) ensures positive experiences throughout all aspects of the customer's journey — from purchase, to first use, to experienced usage, to upgrades, and even through replacement and decommissioning. [Product Management](#) defines the minimal expected product a customer would purchase and the augmented and potential product that differentiates their offering and attracts future customers. The solution's roadmap forecasts the augmented and potential product features over time.

Figure 6. Whole Product Model

Figure 6. Whole Product Model

Solutions Evolve to Meet Changing Needs

To meet the changing needs of customers and markets, teams must be able to deliver new features quickly, receive feedback, and adjust. A fast, automated [Continuous Integration](#) and [Continuous Deployment](#) (CI/CD) pipeline provides developers quick feedback on small changes in the development and operations environmental environment. The development environment offers a close, realistic proxy of the operational environment, as shown in Figure 7.

Figure 7. Get faster feedback from development and operational environments

The solution's vision, backlog, and roadmap include the work to build the CI/CD pipeline. Continuous integration in the development environment provides an economic proxy of the operational environment that can test small changes quickly. However, actual value and feedback can only be evaluated in the operational environment.

Evolving live solutions requires additional constraints. The solution context must provide the ability to deploy changes into live systems and return information from the operational environment. The solution's design must support the collection of user behavior and the operational environment required for feedback. The solution's backlog and roadmap also track and forecast this work.

Solutions are Sustainable

Solutions progress through predictable stages, known as the product lifecycle: introduction to growth, growth to maturity, and maturity to decline (Figure 8). Solutions must frequently evolve to move through these stages and address new market segments and customer needs.

Figure 7. Solutions follow a known product lifecycle

Figure 8. Solutions follow a known product lifecycle

The product lifecycle shows why a Lean-Agile approach to developing solutions is critical. Early solutions are Minimal Viable Products (MVP) [3] that generate validated learning to prove or disprove the solution's business hypothesis (see SAFe's Lean Startup Cycle in the [Epic](#) article). Fast, frequent releases provide new knowledge about the solution's user, market, and technical decisions to refine its backlog, roadmap, and, occasionally, vision.

As described earlier, the Continuous Delivery Pipeline (CDP) enables the frequent, cost-effective change needed to evolve the solution over its lifecycle.

Learn More

[1] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of*

Digital Disruption with the Flow Framework. IT Revolution Press, 2018.

[2] Theodore Levitt. *Marketing Success Through Differentiation* of Anything, Harvard Business Review, 1980, <https://hbr.org/1980/01/marketing-success-through-differentiation-of-anything>

[3] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* Random House, Inc, 2011.

Last Update: 17 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Base milestones on
objective evaluation
of working systems.*

ÑLean-Agile Principle
#5

System Demo

The System Demo provides stakeholders an integrated view of new features for the most recent iteration delivered by all the teams on the ART. Each demo provides an objective measure of progress and the opportunity to give feedback.

Adjust Cookies Settings

A system demo is a critical event. It's the method for assessing the Solution's current state and gathering immediate, [Agile Release Train](#)-level feedback from the people doing the work and critical feedback from [Business Owners](#), sponsors, stakeholders, and [Customers](#). The demo is the one objective measure of value, velocity, and progress of the fully integrated work across all the teams.

Planning for and presenting a useful system demo requires work and preparation by the teams. But it's the only way to get the fast feedback needed to build the right solution.

Details

The system demo tests and evaluates the complete solution in a production-like context (often a staging environment) to receive feedback from stakeholders.

These stakeholders include [Business Owners](#), executive sponsors, other [Agile Teams](#), development management, and customers (and their proxies) who provide input on the fitness for purpose for the solution under development. The feedback is critical, as only they can guide the ART to stay on course or make adjustments.

Figure 1. The System Demo

The system demo occurs at the end of every [Iteration](#). It provides an integrated, comprehensive view of the new [Features](#) delivered by the ART over the past iteration. The system demo offers the ART a fact-based measure of current, system-level progress within the [PI](#). It's the true measure of ART velocity and progress. Achieving this requires implementing the scalable engineering practices necessary to support [Continuous Integration](#) across the ART.

At the end of each PI, the ART holds a final PI system demo that shows all the

features developed over the last PI. Since its scope is larger, the audience may be broader and include customers, [Portfolio](#) representatives, and other additional stakeholders. This demo is usually part of the [Inspect and Adapt \(I&A\)](#) event, which feeds into the retrospective and various PI progress metrics, including the ÔART Predictability MeasureÕ (see [Measure and Grow](#)).

In large [Solution Trains](#), the system demo feeds into the [Solution Demo](#).

The Timing of the System Demo

The system demo takes place as close to the end of the iteration as possible—ideally, the next day. While that is the goal, some complications can make that timing impractical. Immature continuous integration and [Built-in Quality](#) practices can delay the ART’s ability to integrate frequently. Also, each new increment may require extensions to the demo environment, including new interfaces, third-party components, simulation tools, and other environmental assets. While the [System Team](#) strives to provide the proper demo environment at the end of each iteration, the integration may lag.

The system demo must occur within the time bounds of the following iteration. ARTs must make all the necessary investments to allow the system demo to happen in a timely cadence. A lagging system demo is often an indicator of larger problems within the ART, such as continuous integration maturity or System Team capacity.

Balancing Integration Effort and Feedback

The goal of the system demo is to learn from the most recent development experience and adjust the course of action. However, due to costs or availability, some components don’t lend themselves to continuous integration – hardware, mechanical systems, supplier-provided components, and scarce components. Continuous integration may not be economical or practical in such environments.

However, deferred integration, or none at all, is far worse. It significantly inhibits learning and creates a false sense of security and velocity. Therefore, if this is not practical, it’s critical to find the right balance and continuously improve integration and testing automation to lower the cost of future integrations. Figure 2 shows a Ôu-curveÕ cost optimization for integration efforts.

Figure 2. Integration u-curve cost optimization

Figure 2. Integration u-curve cost optimization

When full integration at every iteration is too costly, the teams should consider the following:

Using Test Doubles to speed integration and testing by substituting slow or expensive components with faster, cheaper proxies

Integrating a subset of [Capabilities](#), components, or subsystems

Integrating to illustrate a particular feature, capability, or [Nonfunctional Requirement \(NFR\)](#)

Partial integration with the support of prototypes and mock-ups in place of scarce or expensive components

Less frequent integration (for example, every other iteration) until it's feasible to do it more often

It's also important to remember that continuous integration represents a natural challenge for groups still transitioning to Lean and Agile methods. That's normal and should not be an excuse to reduce the scope or extent of integration. Most of the challenges should disappear as the ART matures.

Continuous integration validated by the system demo contributes to the ability of the enterprise to achieve faster time-to-market through a more continuous flow of value to its customers as outlined in the [Agile Product Delivery](#) competency.

System Demo Attendees

Attendees typically include:

[Product Managers](#) and [Product Owners](#), who are usually responsible for running the demo

One or more members of the System Team, who often help set up the demo in the staging environment

Business Owners, executive sponsors, customers, and customer proxies

[System Architect](#), IT operations, and other development participants

ART Agile team members attend whenever possible

Event Agenda

Having a set agenda and fixed timebox helps lower the transaction costs of the system demo. A sample agenda follows:

Briefly review the business context and the [PI Objectives](#)

Briefly describe each new feature before demoing

Demo each new feature in an end-to-end use case

Identify current risks and impediments

Open discussion of questions and feedback

Wrap up by summarizing progress, feedback, and action items

Guidelines

Here are a few tips for a successful demo:

Timebox the demo to no more than one hour. A short timebox is critical to keep key stakeholders' continuous, biweekly involvement. It also illustrates team professionalism and system readiness.

Share demo responsibilities among the team leads, Product Owners, and even team members who have new features to demo

Demo from the staging environment

Minimize demo preparation. Demo the working, tested capabilities, not slideware.

Minimize demo presentation time. Demo screen snapshots and pictures where appropriate

Discuss the impact of the current solution on NFRs

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.

Last update: 6 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Without involvement, there is no commitment. Mark it down, asterisk it, circle it, underline it. No involvement, no commitment. [1]

ÑStephen Covey

Participatory Budgeting

Adjust Cookie Settings

Participatory Budgeting (PB) is a collaborative process for allocating the portfolio

budget to its value streams.

The [Enterprise](#) provides a portion of its total budget to each portfolio. In turn, [Lean Portfolio Management \(LPM\)](#) allocates the portfolio [Budget](#) to individual [Value Streams](#). The value streams fund the people and resources needed to achieve the current [Portfolio Vision](#) and [Roadmap](#). Empowered Agile Release Trains (ART) advance [Solutions](#) and implement [Epics](#) approved by LPM.

LPM establishes [Lean Budget Guardrails](#) to provide the right mix of investments to address both near-term opportunities and long-term strategy. These guardrails also ensure that significant investments are approved and that the appropriate investment level is allocated to technology, infrastructure, and maintenance. These guardrails and [KPIs](#) promote decentralized decision-making and execution while providing the necessary oversight.

Details

Participatory budgeting (PB) is a dynamic, collaborative process that enables LPM to gather the data and build the consensus required to invest in the best possible solutions. It's a critical element of LPM and is used to establish Lean value stream budgets. PB engages a diverse group of business and technical leaders and other stakeholders in the decision-making necessary to establish and adjust value stream budgets on a regular cadence.

Applying PB to establish SAFe Lean budgets has several benefits:

- Allows the portfolio to adjust budgets to support rapidly changing customer and market needs
- Provides leaders with insights and perspectives from multiple stakeholders about existing solutions and proposed epics
- Creates alignment and buy-in on difficult funding choices, improving employee engagement and morale
- Increases ownership of budgets and results in more realistic and achievable budgets than those imposed top-down
- Improves information sharing and knowledge between leaders and teams

PB is a significant event that requires preparation, coordination, and communication, as illustrated in Figure 1.

Figure 1. Overview of the participatory budgeting process

This article describes how PB is applied in SAFe using an example loan origination and processing portfolio (Figure 2). Its two development value streams (loan application and core banking) manage four solutions, each developed and maintained by a dedicated ART (Figure 2).

Figure 2. Loan application and processing portfolio

Figure 2. Loan application and processing portfolio

1. Preparing the Content

Business Context

The first step in *preparing the content* is establishing and sharing the *business context*. This context typically includes the current state of the business, [Strategic Themes](#), and [Portfolio Vision](#). This information is critical in preparing the investment opportunities to bring into the PB event and guiding the participant's decisions.

Additional information, such as solution and portfolio roadmaps, can help highlight market rhythms and milestones. In our example, the business faces an economic downturn, and LPM will reduce the total portfolio budget allocated for this period.

Total Portfolio Budget and Guardrails

The total portfolio budget is a critical input to the PB process. It describes the money available to fund the value streams within a portfolio. The Lean budgets and guardrails guide how the value streams spend money and influence the mix of investment opportunities brought into the PB event.

The portfolio in our example is associated with a business facing an economic downturn. Accordingly, the enterprise has reduced the total portfolio budget from 24,000 to 21,000. This decrease will require reducing funding to some of the portfolio's value streams.

Baseline Solution Investments

Stakeholders need to understand the current investment context to budget responsibly. Each value stream budget has two parts, as illustrated in Figure 3. The first part, Baseline Solution Investments (BSIs), are the costs to develop, support, and operate a solution that delivers current business capabilities [2]. The second part will be described in the following section, *Proposed Solution Investments*.

BSIs are reviewed during the PB event to ensure existing solutions are funded to meet the business's needs or identified for decommissioning. This budget includes the cost of ongoing development and maintenance of the solution. In some contexts, these costs may include operations, support, sales, and marketing directly applicable to a specific product or service or allocated to a solution as a percentage of the total portfolio costs.

Figure 3. Two components of a value stream budget

Figure 3. Two components of a value stream budget

Typical BSI costs include:

- Employee salary and corporate overhead
- Contractors (such as outsourced development)
- Suppliers (such as solutions used or integrated during development)
- Hardware or other physical materials, licensing, and additional resource costs.
- Corporate allocations for shared services
- Other line items required by the portfolio

LPM can determine BSIs using a simple spreadsheet (Figure 4). The % Allocation column reflects the percentage of the value stream costs allocated to a specific solution, offering a way to analyze these costs. When an ART creates a single solution, the BSI is straightforward; it's merely the ART's cost. When an ART develops more than one, LPM should allocate the value stream costs proportionally to each solution. A portfolio may choose to decompose BSIs into more specific line items to identify potential areas of waste. LPM allocates BSIs for a particular budgeting period (usually twice per year). For example, if the PI

cadence is every three months, then a typical budgeting horizon is every two PIs.

Figure 4. Annual and per PI costs for the credit scoring solution

Figure 4. Annual and per PI costs for the credit scoring solution

Figure 5 illustrates the BSIs for our example portfolio. It shows the annualized solution costs and the costs for the last two PIs and informs the requested budget for the next two PIs—budgeting period.

Figure 5. Baseline solution investments for the loan application and processing portfolio

Proposed Solution Initiatives

Proposed Solution Initiatives (PSIs) describe significant development initiatives needed to improve current solutions or introduce new ones. These may include:

- Portfolio epics (business or enablers)
- Solution epics (which are above the portfolio epic threshold)

Solution epics *below* the portfolio epic threshold are captured as *BSIs*, while those *above* the threshold are captured as *PSIs*.

Business agility requires that the portfolio make the best possible investment decisions for both in-progress and new epics (Figure 6). Accordingly, in-progress epics whose remaining costs are *above* the threshold are included in the PSIs to ensure continued investment is justified, given current market conditions and learning. If new or better opportunities arise, the investment in in-progress epics

can be reduced or stopped. Previous investments from a prior budgeting period are sunk costs and ignored by LPM during decision-making.

Figure 6. Overview of the PSIs for the loan application and processing portfolio

Figure 6. Overview of the PSIs for the loan application and processing portfolio

Establish Budget Time Horizons

Any budgeting process must establish the time horizon of the budgeting period. As noted earlier, value stream budgets are typically updated twice annually (about every two PIs). Since epics typically span PIs, it rarely fits within a specific budgeting period (Figure 7). Accordingly, each PB event should only allocate a portion of the epic's cost likely to be incurred in the upcoming budgeting period.

For an epic that has not yet started, it generally doesn't make sense to budget for the entire estimated implementation cost until the MVP hypothesis has been proven. It may or may not be proven; even if it is, some of the planned investment may continue outside the current budget period. As development proceeds, the estimated implementation cost is refined based on learning from the MVP.

Accordingly, LPM only budgets the epic's investment expected to be incurred in the upcoming budget.

For example, "epic 2" spans two budgeting horizons. The first part of the epic will be included in the current PB event, while stakeholders will consider the remaining cost of "epic 2" in the following PB event. Epic 1 fits within the upcoming budgeting period. However, the in-progress Enabler epic will continue into the following PI, and its remaining cost will also be included in the forthcoming budgeting period.

Figure 7. The budget for a period includes all the work on epics during the period

Determine Proportional Epic Costs for Each Value Stream

Epics typically split across value streams, such as the "Move to Flutter" epic, are funded proportionally by the responsible value streams. In our example, this epic's forecasted cost is 800K. If the core banking solution only requires 200K, then the remaining 600K must be budgeted by the other value stream, which does

the rest of that work. The cost allocations for PSIs should be transparent during PB and enable value stream budgets to be calculated following the event, as shown in Figure 8.

Figure 8. Epic cost inputs for the loan application and processing portfolio

Figure 8. Epic cost inputs for the loan application and processing portfolio

2. Assemble the Participants

PB is most effective when diverse and responsible representatives are involved. Each role brings a unique and valuable perspective to the decision-making process. Attendees typically include:

- LPM
- Product and Solution Management
- Epic Owners
- Enterprise and Solution Architects
- Business Owners

Other relevant stakeholders, such as finance, marketing, or sales

The total number of participants is typically a function of the portfolio's size. Large portfolios require more participants. Since negotiations often break down with more people, SAFe recommends creating groups of five to eight participants for the PB event. To promote healthy debate and learning, each group should have a mix of roles from different value streams. Our example PB event includes 40 people, organized into five groups of eight people.

3. Conduct the Forums

Standard Agenda

The PB event follows a schedule similar to Figure 9. Descriptions of each item follow.

Figure 9. Participatory budgeting event agenda

Figure 9. Participatory budgeting event agenda

Business context ☰ A senior LPM executive or line-of-business owner typically sets the business context for the PB event, including discussing current business performance, revenue or market share, customer satisfaction measures, and more. It often updates operating plans, organizational developments, strategies, and competitive contexts. The presenter will typically discuss strategic themes and business objectives for the upcoming periods, including a review of the portfolio roadmap, when needed.

Epic and solution briefings ☰ Epic Owners present a brief overview of PSIs, and product management gives an overview of BSIs for each solution and critical market events and milestones.

Participatory budgeting process ☰ A facilitator presented the PB process and expected outcomes of the event.

Participatory budgeting forums ☰ Participants are organized into groups of five to eight people with various roles and people from different Value Streams. Each group gets the total portfolio budget equally divided among the participants. Each group debates and collaboratively funds the BSIs and PSIs.

Moving forward ☰ LPM presents the initial results of participatory budgeting and discusses the next steps.

Participatory budgeting retrospective ☰ A facilitator leads a brief retrospective for the PB event to capture what went well, what didn't, and what can be done better next time.

Participating in a Forum

At the start of the PB forums, each group of participants is given the list of BSIs and PSIs and their forecasted costs (Figure 10). At the forum's start, the group is assigned the total portfolio budget to allocate, and each *participant* gets an *equal portion*.

In our example, the total requested budget for all BSIs and epics is 24,900K, exceeding the 21,000K portfolio budget by 3900K. Since the enterprise has already reduced the portfolio's total budget allocation from 24,000K to 21,000K, surpassing it is not an option. However, LPM could reduce investment in some BSI to ensure funding for the requested epics. Alternatively, the LPM team could stop in-flight epics or defer new epics. However, these complex decisions require expert knowledge and judgment of key business and technology stakeholders. PB's primary purpose is to get this feedback from key stakeholders and determine how to allocate the budget best.

Figure 10. Total forecast for BSIs and PSI for the loan application and processing portfolio

Since no participant can likely fund BSIs and epics independently, the group must collaborate to pool their money and identify and support the best investments. Generally, the first order of business is to invest in BSIs based on the current business context. In other words, the current BSIs may not meet the needs of the evolving strategy, and these must be analyzed and potentially adjusted.

To fund initiatives, individuals in each group must pool their budgets. Because partially funded solutions and epics are candidates for termination or cutbacks, the group will negotiate to determine where to make the best investments. The discussions from this collaboration allow participants to make choices that optimize value delivery across the portfolio rather than focusing on their value streams.

After funding or adjusting the BSI, teams are encouraged to support their highest-priority epics fully. This guidance avoids spreading too little money across too many initiatives that are not funded sufficiently. Full investment in a selected set of

initiatives is the key to a strategy that builds on an enterprise's strengths and creates viable and differentiated product offerings.

During a forum, a group may:

Invest more than the estimate if they have reason to believe the costs are too low or if they think it should be delivered sooner or with more value

Invest less if they have reason to believe the costs are too high or if the investment should be deferred to a future budget period

Figure 11 shows how one group allocated its budget during the PB forum.
Highlights include:

It deferred the introduction of new features in the channel management and loan origination solutions, which reduced the value stream's BSI

It fully funded the revenue-generating Facebook login and college student loans

It reduced the scope of investment in the core banking microservice enabler epic

Figure 11. One group's budget recommendation

Figure 11. One group's budget recommendation

4. Analyze Results

The aggregated results of the PB forums are analyzed to identify suggested funding patterns. In Figure 12, for example (derived from an actual PB event using SAFe Collaborate), it is easy to see alignment on funding priorities:

Figure 12. An example of an output of a participatory budgeting event

Figure 12. An example of an output of a participatory budgeting event

While the PB event results do not directly determine the budget allocations for the value streams, they provide critical information and input that LPM needs to make an informed decision that maximizes the portfolio's business outcomes.

Some common patterns of investment include:

Investing a consistent amount in one or more solutions that are generating stable

returns

Reducing investments in one solution to invest more in a different solution with a better-projected return

Investing in the decommissioning of obsolete solutions to free resources for innovation

Investing across all solutions to realize a standard set of business requirements, such as compliance or serving a new or adjacent market

Reducing investments across all solutions during an economic downturn

These patterns are not exhaustive, and a specific portfolio's needs will determine how they manage solutions investments.

Given this input, the LPM team makes final determinations to implement these changes, including:

Finalizing the portfolio budget

Allocating funding to value streams

Verifying that the investments align with the Lean budget guardrails

Engaging with HR, finance, legal, operations, and any other affected groups

Value streams typically prepare for these changes during the current PI and implement them at the start of the following PI.

Adjusting Value Streams Dynamically within Approved Budgets

After LPM finalizes budgets based on the analysis, the value streams adjust their work dynamically to address current facts, local context, and emerging business results. Some examples may include the following:

If the epic hypothesis is proven, the value stream will continue to work on the epic, using the allocated budget.

If it's not proven, the value stream will stop working on the epic or pivot to a new hypothesis and epic. In the meantime, empowered ARTs pull the next approved epic from the portfolio backlog or the next set of approved work items from the ART backlog.

This dynamic structure is the essence of Lean budgeting and provides a far more Agile way to address budget changes and have the teams work on what matters at that time.

Learn More

[1] <https://www.goodreads.com/quotes/133327-without-involvement-there-is-no-commitment-mark-it-down-asterisk>

[2] BSIs are the equivalent of ÔRun the BusinessÕ (RTB) or ÔBusiness as UsualÕ (BAU) costs, but for a specific solution.

Last updated: 5 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
 Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Unless commitment
is made, there are
only promises and
hopes; but no plans.*

ÑPeter F. Drucker

Planning Interval (PI)

A Planning Interval (PI) is a cadence-based timebox in which Agile Release Trains deliver continuous value to customers in alignment with PI Objectives.

Adjust Cookies Settings
PIs are typically 8 – 12 weeks long. The most common pattern for a PI is four development Iterations, followed by one Innovation and Planning (IP) Iteration. The

PI is for an ART like an [Iteration](#) is for [Agile Teams](#). It's a fixed timebox for planning, building, validating, delivering value, and getting fast feedback.

During the PI, Agile Teams apply cadence and synchronization to combine the work of multiple teams into one or more releasable increments. Individual teams may also release value independently, depending on context. The cadence and synchronization of the PI enable the ART to:

- Plan the ART's next increment of work
- Limit work in process (WIP)
- Summarize newsworthy value for feedback
- Ensure consistent ART retrospectives

Due to its broad scope, the PI provides an appropriate timebox for [Portfolio](#) considerations and road mapping.

Details

SAFe divides the development timeline into a series of iterations within a PI. The Big Picture illustrates how a PI begins with a [PI Planning](#) event and is followed by four execution iterations, concluding with one IP iteration (Figure 1).

Figure 1. Typical PI timebox

Figure 1. Typical PI timebox

This pattern is suggestive but *arbitrary*, and there is no fixed rule for how many iterations are in a PI. However, experience has shown that a PI duration between 8 and 12 weeks works best, with a bias toward shorter durations. Organizations implementing [Lean Portfolio Management](#) (LPM) often find it beneficial to match the schedule of the *Strategy Portfolio Review* and *Participatory Budgeting* events to improve the alignment of strategy with execution. (Please read the events section in the LPM competency article for more information.)

Develop on Cadence

PIs provide the development rhythm for trains and the assets they build to grow iteratively and incrementally. “Develop on cadence” is a label for a coordinated set of practices that support Agile Teams by offering a reliable series of events and activities on a regular, predictable schedule. However, the planning cadence is often different from the release cadence. This approach helps customer-centric enterprises create a continuous value flow for their customers.

The business determines the timing of releases depending on market and customer needs and the organization's motivation to provide value. Some enterprises frequently release during the PI, while others may be constrained by compliance, supplier dependencies, or other business and market requirements, driving less frequent releases. Decoupling the development events and activities that support value creation from how and when that value is delivered further promotes [Business Agility](#).

ART Events Drive Development

When it comes to execution for a single ART, a dual Plan-Do-Check-Adjust (PDCA) sequence of events creates a closed-loop system to keep the train on the tracks, as illustrated in Figure 2. The *outer loop* represents the *ART's PDCA cycle for the PI*, while the inner circle represents an Agile Team's PDCA cycle for an *iteration*. In this example, the team is using Scrum for the inner loop. The diagonal line (PI Start) shows that after ART completes PI Planning, the teams on the train start the inner PDCA loop.

Figure 2. ART events drive development cadence

Figure 2. ART events drive development cadence

The following sections describe each ART event. The [SAFe Scrum](#) article offers guidance for the iteration events shown in Figure 2, and [SAFe Team Kanban](#) describes the cycle for the inner loop when using this method.

Plan the PI

Each ART PI begins with a [PI Planning](#) event. During this event, the teams estimate what will be delivered and highlight their dependencies with other Agile Teams and trains. One primary outcome of PI planning is a set of [PI Objectives](#) detailing what the ART should have to demo at the end of the PI. Moreover, Agile Teams continuously integrate their work throughout the PI and demo functionality during the [Iteration Review](#) and [System Demos](#).

PI Planning has a standard agenda that includes a presentation of business context and [vision](#), followed by team planning breakouts—where each team creates its [Iteration](#) plans and objectives for the upcoming [PI](#). Facilitated by the [Release Train Engineer \(RTE\)](#), PI planning includes all members of the ART and occurs within the [Innovation and Planning \(IP\) Iteration](#).

Conduct System Demos

The system demo event tests and evaluates the solution in a production-like environment (often staging) to receive feedback from stakeholders, including [Business Owners](#), executive sponsors, other Agile Teams, development management, and customers (or their proxies). This stakeholder feedback is critical, as only they can evaluate the effectiveness and usability of the solution under development and offer the guidance the ART needs to stay on course or adjust. The system demo event occurs at the end of every [Iteration](#), providing an integrated, comprehensive view of the new [Features](#) delivered by the ART.

Inspect and Adapt

Each PI concludes with an [Inspect and Adapt](#) (I&A) event, a regular time to reflect, apply problem-solving techniques, and identify improvement actions needed to increase the following PI's velocity, quality, and reliability. During the I&A, Product Management or other team members showcase all the finished features during the final PI system demo. Quantitative and qualitative measurements and a retrospective problem-solving workshop follow the demo. The result of the I&A is a set of improvement features or stories that the RTE or teams can add to the backlog for the upcoming PI planning. In this way, every ART improves every PI, including course corrections to the solution.

Iterations Drive ART Cadence

All Agile Teams contribute to the ART's Iterations. However, the specifics of planning and execution during the iteration may differ based on whether the team works in Scrum or Kanban. Those who apply Team Kanban typically work in a continuous flow model, contributing to the ART's increment during each iteration.

Since Agile teams operate as part of an ART, their cooperation is critical for meeting the agreed-to [Team and ART PI Objectives](#). Accomplishing this collaboration requires teams to align to the same iteration cadence and duration.

Deliver Continuously and Release on Demand

Building and maintaining a [Continuous Delivery Pipeline](#) (CDP) allows each ART to define, build, validate, and release new functionality to meet their PI objectives. Multiple Agile teams on the ART share the same CDP as they collaborate within iterations throughout the PI. Each aspect of the CDP is described below:

Note: For some ARTs, continuous delivery means releasing multiple times per day. For others, "continuous" means releasing weekly or monthly—or whatever cycle satisfies market demands and the goals of the enterprise.

Continuously Exploration (CE)

[Continuous Exploration](#) focuses on creating alignment on what needs to be built. In CE, design thinking ensures the enterprise understands the market problem, customer needs, and the solution required to meet that need. It starts with an idea or a hypothesis of something that will provide value to customers, typically in response to customer feedback or market research. The concept is further analyzed and researched to understand the requirements for a Minimum Marketable Feature (MMF) or Minimum Viable Product (MVP). Finally, convergence occurs by understanding which features will likely meet customer and market needs.

Continuously Integration (CI)

[Continuous Integration](#) focuses on taking features from the ART backlog, applying design thinking tools in the problem space to refine features, conducting user research, and collecting feedback. After they are clearly understood, Agile Teams implement them. Completed work is committed to version control, built and integrated into a system or solution increment, and tested end-to-end before

validating it in a staging environment.

Continuously Deployment (CD)

[Continuous Deployment](#) takes the changes from the staging environment and deploys them to production. At that point, they are verified and monitored to ensure they are working correctly. This step makes the features available in production, where the business determines the appropriate time to release them to customers. This aspect allows the organization to respond, roll back, or fix forward.

Release on Demand (RoD)

[Release on Demand](#) makes value available to customers all at once or incrementally based on market and business needs. RoD allows the business to release when market timing is optimal and carefully manages the risk associated with each release.

Manage Flow, Scope, Risk, and Dependencies

When viewed as a whole, continuous delivery is an extensive process. Indeed, it may be the most vital capability of every ART. Stakeholders need to visualize and track the ongoing work, even when a significant portion of the pipeline is automated. The ART requires establishing WIP limits to improve throughput and identify and address bottlenecks. That's the role of the ART Kanban. In addition, Kanban sync events manage flow, scope, risk, and dependencies, as described in the following sections.

Visualize and Limit WIP with the ART Kanban

The RTE, Product Management, and others use the ART Kanban to visualize, track and manage the flow of features from ideation to analysis, implementation, and release through the [Continuous Delivery Pipeline](#). Figure 3 illustrates a typical ART Kanban with example policies governing the entry and exit of features in each process state. It helps the ART improve flow by matching demand to capacity, applying WIP limits, visualizing bottlenecks, and identifying opportunities for relentless improvement. Product Management reviews the Kanban and pulls in more work, respecting WIP limits in collaboration with Product Owners. This Kanban also facilitates reviewing and prioritizing new features based on continuous exploration and offering the information needed to make release and deployment decisions.

Figure 3. ART Kanban board

Figure 3. ART Kanban board

Sync on Scope, Progress, and Dependencies

Three sync events help the ART stay on track, as illustrated in Figure 4. The *Coach Sync* focuses on executing the current PI, including risk, dependencies, progress, and impediments. The *PO Sync* manages the PI's scope, reviews progress, adjusts priorities, and prepares for the following PI. Since Product Owners and Scrum Masters/Team Coaches are often interested in similar topics and need to collaborate, sometimes it's helpful to combine the Coach Sync and PO Sync into a single event known as an ART Sync. The ART Sync usually replaces the Coach Sync and PO Sync for a particular iteration to reduce overhead.

Figure 4. Example PO Sync agenda

Figure 4. Example PO Sync agenda

Address Risks with the ROAM Board

The ROAM board created during PI planning (Figure 5) can be reviewed during the ART Sync to ensure those responsible for owning or mitigating a risk take the necessary actions. The ART may also record any new items that may have arisen after planning and ROAM them.

Figure 5. Example ROAM Board

Figure 5. Example ROAM Board

Resolve Dependencies with the ART Planning Board

During the sync events, the RTE, Product Management, Scrum Masters/Team Coaches, and other stakeholders can use the ART Planning Board, shown in Figure 6, to track and manage dependencies, ensuring they do not block other teams.

Figure 6. ART Planning Board

Figure 6. ART Planning Board

The ART planning board should be used during and after PI planning to help identify ways to reduce or eliminate dependencies, enabling teams to work more independently and increase value flow.

Last update: 7 February 2023

images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The great thing about fact-based decisions is that they overrule the hierarchy.

~ Jeff Bezos

Measure and Grow

Measure and Grow is an approach SAFe enterprises use to evaluate progress towards Business Agility and determine improvement actions.

[Adjust Cookie Settings](#)

Business Agility sets new performance standards for organizations, requiring fast,

effective responses to emerging business opportunities. However, to improve speed and agility, leaders, teams, and business stakeholders need a way to reliably measure the current state and identify what they can do to improve. Therefore, choosing what and how to measure is a critical enabler of continuously improving business performance. This article describes a comprehensive approach that can be used to measure the performance of a SAFe portfolio or any of its elements.

Details

When it comes to metrics, the first and most important thing is understanding *what* to measure. The goal of *Business Agility* is clear: quickly respond to market changes and emerging opportunities with innovative, digitally-enabled business solutions. The *Business Agility Value Stream*, shown in Figure 1, visualizes the steps needed to achieve this. SAFe's three measurement domains, *Outcomes*, *Flow*, and *Competency*, support this process and provide a comprehensive yet simple model for measuring progress toward this goal. The insights provided by these three measurement domains support better decision-making and help to identify opportunities for improvement.

Figure 1. Three SAFe measurement domains support the goal of business agility

Figure 1. Three SAFe measurement domains support the goal of business agility

The three measurement domains are defined as follows:

Outcomes: Do our solutions meet the needs of our customers and the business?

Flow: How efficient is the organization at delivering value to the customer?

Competency: How proficient is the organization in the practices that enable business agility?

Furthermore, these three measurement domains are applicable at every level of an organization. As Figure 2 illustrates, they can be used to measure performance within a SAFe portfolio, a Solution Train, an Agile Release Train, or even a single Agile Team.

Figure 2. The three measurement domains are applicable at all levels of a SAFe Enterprise

Each measurement domain contains a set of specific metrics, which are described in the sections below.

Measuring Outcomes

Outcomes help determine whether a development organization's efforts produce the desired business benefit. Outcomes may measure externally facing concerns such as increases in revenue, customer retention, etc., as well as internal considerations such as employee engagement.

KPIs and OKRs

A SAFe Portfolio measures outcomes using *Key Performance Indicators (KPIs)* and *Strategic Themes*. Each KPI is a specific and quantifiable measure of business results for the value streams within that portfolio. Outcome metrics of this kind are typically context-specific and depend heavily on the organization, business model, and the nature of solutions delivered to the customer. For example, the customer conversion rate may be a meaningful metric for an eCommerce business but would be inapplicable to a microchip manufacturer. Some indicators, however, may be successfully applied across contexts, such as Net Promoter Score, for example.

The [Value Stream KPIs](#) article provides guidance for defining appropriate KPIs for that particular SAFe Portfolio. Examples of KPIs appear in Figure 3.

Figure 3. Value stream KPIs are specific to the type of operational value stream supported

Of course, these KPIs are partly informed by the portfolio strategic themes since the strategy helps to determine the targets to be met. However, whereas KPIs represent ongoing ÔhealthÕ metrics that can be used to measure overall business performance, the strategic themes, formulated as OKRs, define the specific outcomes that the portfolio is working towards to achieve future success. Therefore the key results associated with these objectives determine another set of critical outcome metrics that are typically measured quarterly, as shown in Figure 4 below.

Figure 5. Measuring progress of strategic themes using OKR metrics

Figure 4. Measuring progress of strategic themes using OKR metrics

Within a large portfolio, it can be useful to create specific OKRs for each value stream that align with the portfolio strategic themes. And further, for large value streams that contain multiple ARTs, this process can be repeated to create a set of OKRs that define the goals for each specific ART. This approach also allows those at each level of the organization to see the direct impact of their work against the key results of the OKRs they are aligning to. (This use case, and others, are described in more detail in the [OKRs article](#))

Employee Engagement

Another important internal outcome metric is employee engagement. Employee engagement measures the amount to which individuals feel motivated and actively engaged in supporting the achievement of the organization's goals and values. Higher levels of employee engagement result in higher productivity, efficiency, and innovation levels. Consequently, lower levels of employee engagement can lead to poor motivation, lower-quality work, and higher staff turnover.

Different methods exist for measuring employee engagement, and each organization needs to determine what is right for them. Some organizations will use an annual employee engagement survey. Others use an employee Net Promoter Score (eNPS), which asks, "How likely are you to recommend your employer to others as a place of work?" and is measured on a 10-point scale. Whichever approach is chosen, the resultant data should inform initiatives to improve employee engagement levels.

Iteration Goals and PI Objectives

Localized metrics such as [Iteration Goals](#) and [PI Objectives](#) are used effectively by teams and trains to measure whether they are achieving their outcomes. These ensure their efforts are focused on the needs of the customer and the business, provide feedback on the progress they are making toward business results, inform the prioritization process, and facilitate acceptance of work.

Establishing effective outcome metrics requires a close collaboration of trains, value streams, and portfolios with their business partners, who can best define the business benefits resulting from solution investment.

Measuring Flow

Flow measures are used to determine how effective an organization is at delivering value. The Flow Framework created by Mik Kersten [1] provides five metrics that can be used to measure different aspects of flow. As SAFe is a flow-based system, each metric is directly applicable. In addition, SAFe defines *Flow Predictability* to measure how Teams, ARTs, and Solution Trains deliver business value against their planned objectives. These six flow metrics are shown in Figure 5 and described further below.

Figure 5. The six SAFe flow metrics

Figure 5. The six SAFe flow metrics

Flow Distribution

What does it measure? Flow distribution measures the amount of each type of work in the system over time. This could include the balance of new business [Features](#) (or [Stories](#), [Capabilities](#), or [Epics](#)) relative to [Enabler](#) work, as well as the work to resolve defects and mitigate risks. Alternatively, a helpful view of portfolio flow distribution might illustrate the distribution of funding allocation across investment horizons.

How is this measured? One simple comparison is to count the number of each type of work item at any point in time. A more accurate measure might consider the size of each work item. Agile Teams may measure flow distribution per iteration, but PI boundaries are commonly used to calculate this at the ART level and above, as shown in Figure 6.

Why is this important? To balance both current and future velocity, it is important to be able to track the amount of work of each type that is moving

through the system. Too much focus on new business features will leave little capacity for architecture/infrastructure work that addresses various forms of technical debt and enables future value. Alternatively, too much investment in technical debt could leave insufficient capacity for delivering new and current value to the customers. Target capacity allocations for each work type can then be determined to help balance these concerns. Returning to the portfolio example, tracking the distribution of funding across investment horizons provides a means to ensure a balanced portfolio that ensures both near- and long-term health.

Figure 5. Flow distribution over time

Figure 6. Flow distribution over time

Flow Velocity

What does it measure? Flow velocity measures the number of backlog items (stories, features, capabilities, epics) completed in a given timeframe; this is also known as the system's throughput. (Figure 7).

How is this measured? As with flow distribution, the simplest measure of velocity

is to count the number of work items completed over a time period such as an iteration or PI. Those items can be stories, features, capabilities, or even epics. However, since work items are not all the same size, a more common measure is the total number of completed story points for work items of a type over the timeframe.

Why is this important? All other things being equal, higher velocity implies a higher output and is a good indicator that process improvements are being applied to identify and remove delays from the system. However, the system's velocity will not increase forever, and over time stability of the system is important. Significant drops in velocity highlight problems that warrant investigation.

Figure 7. An example of an Agile team's flow velocity in story points per iteration

Flow Time

What does it measure? Flow time measures the total time elapsed for all the steps in a workflow and is, therefore, a measure of the efficiency of the entire

system. Flow Time is typically measured from ideation to production. Still, it can also be useful to measure Flow Time for specific parts of a workflow, such as code commit to deployment, to identify opportunities for improvement.

How is this measured? Flow time is typically measured by the average length of time it takes to complete a particular type of work item (stories, features, capabilities, epics). A histogram is a useful visualization of flow time (Figure 8) since it helps identify outliers that may need attention and supports the goal of reducing the overall average flow time.

Why is this important? Flow time ensures that organizations and teams focus on what is essential → delivering value to the business and customer in the shortest possible time. The shorter the flow time, the less time our customers spend waiting for new features and the lower the cost of delay incurred by the organization.

Figure 7. Measuring feature flow time with a histogram

Figure 8. Measuring feature flow time with a histogram

Flow Load

What does it measure? Flow load indicates how many items are currently in the system. Keeping a healthy, limited number of active items (limiting work in process) is critical to enabling a fast flow of items through the system ([SAFe Principle #6](#)).

How is it measured? A Cumulative Flow Diagram (CFD) is one common tool used to effectively visualize flow load over time (Figure 9). The CFD shows the quantity of work in a given state, the rate at which items are accepted into the work queue (arrival curve), and the rate at which they are completed (departure curve). At a given point in time, the flow load is the vertical distance between the curves.

Figure 8. Visualizing flow load with a cumulative flow diagram.

Figure 9. Visualizing flow load with a cumulative flow diagram.

Why is this important? Increasing flow load is often a leading indicator of excess work in process. All other things being equal, the likely result will be an increase in future flow times as queues start to build up in the system. For this reason,

measuring and reducing flow load is of critical importance. Furthermore, it is easy to see how more frequent delivery lowers flow load while improving flow time and flow velocity.

Flow Efficiency

What does it measure? Flow efficiency measures how much of the overall flow time is spent in value-added work activities vs. waiting between steps.

How is it measured? To correctly measure flow efficiency, the teams, trains, and value streams must clearly understand what the flow is in their case and what steps it passes through. This understanding is achieved with the help of Value Stream Mapping – a process of identifying workflow steps and delays in a system, as shown in Figure 6. (For more on Value Stream Mapping, see the [Continuous Delivery Pipeline](#) article and [2]. In addition, the [SAFe DevOps course](#) provides comprehensive guidance on performing Value Stream Mapping.) Once the steps have been mapped, flow efficiency is calculated by dividing the total active time by the flow time and is expressed as a percentage, as shown in Figure 10.

Why is this important? In a typical system that has not yet been optimized, flow efficiency can be extremely low, often in single digits. A low flow efficiency highlights a lot of waste in the system, along with bottlenecks and delays that should be addressed. Conversely, the higher the flow efficiency, the better the system can deliver value quickly.



Figure 10. Flow efficiency is the ratio of total active time to total flow time (this one is too efficient)

Flow Predictability

What does it measure? Flow predictability measures how well teams, ARTs, and Solution Trains can plan and meet their PI objectives.

How is it measured? Flow Predictability is measured via the ART Predictability Measure, Figure 11. This measure calculates the ratio of planned business value achieved to actual business value delivered in a PI. For more information on calculating this important metric, see the [Inspect and Adapt](#) article.

Why is this important? Low or erratic predictability makes delivery commitments unrealistic and often highlights underlying problems in technology, planning, or organization performance that need addressing. Reliable trains should operate in the 80 Ð 100 percent range; this allows the business and its stakeholders to plan effectively.

Figure 11. ART predictability measure

Figure 11. ART predictability measure

Note on DORA Metrics: Within and across the three measurement domains, it can often be helpful to bring together complementary metrics to provide a specific view of performance. An example is the DORA metrics used to measure the performance of an organization's DevOps capabilities [3]. The four DORA metrics are 1) deployment frequency, 2) lead time for changes, 3) time to restore service, and 4) change failure rate.

Each of these is an application of a flow metric designed for a particular use case. *Deployment frequency* is a productivity metric and an example of flow velocity. Instead of stories completed per iteration, it measures the number of deployments per given time period. Both *lead time for changes* and *time to restore service* are examples of flow time metrics, focusing on specific steps in the workflow. Finally, *change failure rate* represents the percentage of changes that require remediation after they have gone to production. In other words, how often does the work that arrives at the "deploy to production" step in the workflow contain

errors? When creating a value stream map to measure flow efficiency, this is captured in the percent complete and accurate (%C&A) metric for each step i.e. the percentage of work that the next step can process without needing rework. High rates of change failure contribute significantly to low flow efficiency.

Measuring Competency

Achieving business agility requires a significant degree of expertise across the [Seven SAFe Core Competencies](#). While each competency can deliver value independently, they are also interdependent in that true business agility can be present only when the enterprise achieves a meaningful state of mastery of all.

Measuring the level of organizational competency is accomplished via two separate assessment mechanisms designed for significantly different audiences and different purposes. The SAFe Business Agility Assessment is designed for the business and portfolio stakeholders to assess their overall progress on the ultimate goal of true business agility, as shown in Figure 12.

Figure 11. An example business agility assessment.

Figure 12. A completed business agility assessment.

The spreadsheet version of the assessment can be downloaded here.

[Download the SAFe Business Agility Assessment](#)

Note for SAFe Studio Members: All the SAFe assessments are available for SAFe Studio Members online through our partner Comparative Agility. This provides additional data collection, analysis, comparison, and trending capabilities that can be used to improve performance. Access these from the [Measure and Grow SAFe Studio page](#).

The SAFe Core Competency Assessments help teams and trains improve on the technical and business practices they need to help the portfolio achieve that larger goal. There is one for each of the seven core competencies. The Team and Technical Agility Assessment as an example, in Figure 13.

Figure 12. A report from a team and technical agility competency assessment

Figure 13. A report from a team and technical agility competency assessment

Each assessment follows a standard process pattern of running the assessment, analyzing the results, taking action, and celebrating successes. In addition, comparative analysis against the competition is achievable via online assessment tools available to SAFe community members. Additional information and guidance can be found in the Advanced Topic Article [Successfully Facilitating SAFe Assessments](#).

The following table provides download links for each of the core competency assessments.

Core competency assessments download

[Organizational Agility](#)

[Lean Portfolio Management](#)

Enterprise Solution Delivery

Agile Product Delivery

Team and Technical Agility

Continuous Learning Culture

Lean-Agile Leadership

Measuring and Managing DevOps Maturity

In addition to the business agility and core competency assessments, the SAFe DevOps Health Radar (Figure 14) is an assessment that helps ARTs and Solution Trains optimize their value stream performance. It provides a holistic DevOps health check by assessing the maturity of the four aspects and 16 activities of the continuous delivery pipeline. The Health Radar is used to measure baseline maturity at any point in a DevOps transformation and guide fast, incremental progress thereafter.

Note: The DevOps Health Radar should be used alongside the Agile Product Delivery assessment to ensure full coverage of all three dimensions of the APD core competency.

Figure 14. The SAFe DevOps Health Radar

Figure 14. The SAFe DevOps Health Radar

Download the free DevOps Health Radar assessment [here](#). AgilityHealth also offers an [online version of this assessment](#).

Four Critical Success Factors for Effective Measurement

Measuring organizational performance is one of the most sensitive areas in every business, often subject to politics and various dysfunctions. Additionally, since measurement inevitably involves the interpretation of data, it is subjected to cognitive bias, communication issues, and alignment disconnects. All this leads to a substantial danger in any measurement system: if not correctly implemented, *some measurements can do more harm than good*. The following success factors will help guide the enterprise to more effective measurements and more importantly, better business results.

1. Use measurement in conjunction with other discovery

tools

However well-designed, any measurement system provides only a partial picture of reality, and adding more metrics does not necessarily improve visibility. There is a story behind every number, and that story often contains more important information than the number itself can convey. A powerful tool to be used in conjunction with measurement is *direct observation* (Gemba) – observation of the actual environment where value is created and where it meets the customer. Formal measures and informal observations reinforce one another. But used in isolation, “managing by just the numbers” can lead to poor outcomes and even worse morale.

2. Apply metrics where they support improved decision-making

A common trap when applying metrics is over-measuring for fear of not measuring enough. Although many metrics can be automated, as the number of metrics and frequency of measurement increases, so will the effort needed to collect and analyze the data. When considering whether to include an additional metric in your measurement system it can be prudent to ask the question, “what decisions will this metric help inform that isn’t supported today with our existing metrics?” If the new metric helps to drive better decision-making, it should be a candidate for inclusion; if not, then omit it. A further clarifying question is “do we need to measure this right now?” This question recognizes that the metrics we use will change over time as the decisions we need to make change throughout the development process.

3. Understand the effect of metrics on behaviors

In a positive culture, knowledge workers are motivated to deliver winning solutions and work with purpose, mastery, and autonomy. However, when too much emphasis is placed on a specific numerical indicator, and when that indicator is directly tied to compensation or career growth opportunities, achieving that number becomes the goal instead of creating effective solutions.

Additionally, the pressures to succeed often lead to the misuse of metrics. For example, flow efficiency may be used to assign blame for a missed delivery date to a particular ART that has become a bottleneck rather than using this information to identify systemic problems that need addressing. Perhaps the root cause was a lack of resources or changing priorities outside the ART’s control.

In each case, SAFe’s Core Values of transparency and alignment must provide the proper foundation for an effective measurement system alongside creating an environment where *the facts are always friendly*.

4. Interpret metrics carefully

Just collecting specific measures is not enough. If interpreted without proper understanding, an indicator may be quite misleading. For example, when measuring flow time, the work items must be *actual, valuable* features (stories, and so on) that carry business benefits; otherwise, the train may be reporting improvements in the flow of work but struggling to get any real value out of the door.

Learn More

[1] Kersten, Mik. *Project to Product*. IT Revolution Press, 2018.

[2] Martin, Karen. *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation*. McGraw-Hill Education, 2013.

[3] Accelerate 2022 – State of DevOps Report.

<https://cloud.google.com/devops/state-of-devops>

Last update: 21 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite

[presentations](#)

GET SOCIAL

[Permissions Form](#)

300

[Blog](#)

[Twitter](#)

[Usage and Permissions](#)

Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[Linkedin](#)

BUSINESS

[About Certification](#)

[YouTube](#)

HOURS

[Become a Trainer](#)

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The productivity of software delivery at enterprise organizations falls woefully behind that of the tech giants, and the digital transformations that should be turning the tide are failing to deliver business results.

ÑMik Kersten,
*Project to Product:
How to Survive and Thrive in the Age of*

Adjust Cookie Settings

Business Agility

Business Agility is the ability to compete and thrive in the digital age by quickly responding to market changes and emerging opportunities with innovative, digitally-enabled business solutions.

Everything moves fast in the digital age. Customer desires, competitive threats, technology choices, business expectations, revenue opportunities, and workforce demands now happen at *blistering speeds*.

Today, achieving customer delight at the speed of the market requires validating innovations with customers and then Ôpivoting without mercy or guiltÕ when the hypothesis needs to change. Moreover, significant technological advances are unlocking new ways to create this value. For example, [AI](#), [Big Data](#), [Cloud](#), and [DevOps](#) enable enterprises to expand their product lines, modernize their existing offerings, scale to mass markets, make better fact and insight-based decisions, and streamline solution development.

Competing in the Age of Software

In her book *Technological Revolutions and Financial Capital*, Carlota Perez [2] explains the evolution of business, society, and financial cycles based on her analysis of five significant technological revolutions over the past three centuries. Her research begins with the *Industrial Revolution*, leading to the *Age of Steam and Railways*, the *Age of Steel and Heavy Engineering*, and the present *Age of Software and Digital*, as illustrated in Figure 1.

Perez concludes that these revolutions lead to massive social change, market disruption, and an all-new economic order. Indeed, these are *world-shaking* disruptions that typically occur once in a generation.

Figure 1. Technological revolutions change society

Figure 1. Technological revolutions change society

We are in the midst of one of those ages now, the deployment period of the **age of software and digital**. This period is when every business is a software business. Put simply, competing in this age requires large-scale software and system development capability that enables true business agility.

Why Organizations Struggle to Achieve Business Agility



The organizations we created in the 20th century were designed much more for reliability and efficiency than for agility and speed.

Ñ John P. Kotter [3].

Most leaders in traditional organizations are aware of the digital disruption threat, yet many fail to transition to take their place in the next economy. The question is, why?

Organizations Start as a Fast Adaptive Network

As an organizational researcher and author, John Kotter illustrates in his book, *Accelerate: Building Strategic Agility for a Faster-Moving World* [4], successful enterprises don't start as large and cumbersome. Instead, they typically began as a fast-moving, adaptive network of motivated individuals focused on responding to the customer and the new business opportunity. Roles and reporting relationships are fluid, and people collaborate organically to identify customer needs, explore potential solutions, and deliver value in any way they can. In other words, it's an adaptive entrepreneurial network of people working to leverage an opportunity (Figure 2).

Figure 2. New enterprises operate as networks focused on the customer and new business opportunities

Hierarchy Forms, then Grows and Grows

As the enterprise succeeds, it naturally wants to expand on its success and grow. This growth means that individual responsibilities must become clearer. As a result, the enterprise hires specialists to add expertise, creating new functional areas.

Policies and procedures ensure legal adherence and compliance, driving repeatable, cost-efficient operations. The business starts to organize functionally to scale, causing silos to form. Meanwhile, operating in parallel, the network continues to seek new opportunities to deliver value (Figure 3).

Achieving larger economies of scale requires the hierarchy to grow. And it grows until it conflicts with the entrepreneurial network.

Figure 3. Growing hierarchy running in parallel with an entrepreneurial network

Figure 3. Growing hierarchy running in parallel with an entrepreneurial network

The Hierarchy and Adaptive Network Collide

Eventually, the hierarchy collides with the faster-moving, more adaptive network. The result? The adaptive network gets crushed. The focus on the customer is often one of the main casualties (Figure 4.)

Figure 4. Entrepreneurial network collides with a growing hierarchy

Figure 4. Entrepreneurial network collides with a growing hierarchy

Without the entrepreneurial network, the organization lacks the agility to respond when the customer needs shift dramatically or when a disruptive technology or competitor emerges. An urgent crisis erupts, and the company's survival is now at stake.

However, the organizational hierarchies built over the last fifty years have provided time-tested structures, practices, and policies. They support the recruiting, retention, and growth of thousands of employees across the globe. Simply put, they are still needed.

In addressing this dilemma, Kotter points out, *“The solution is not to trash what we know and start over but instead to reintroduce a more agile, network-like structure”* that operates in concert with the hierarchy to create what he calls a “dual operating system.” This system, illustrated in Figure 5 and described in the following section, allows companies to capitalize on rapid-fire strategic challenges and retain their stability [3].

The Solution: SAFe is the Second Operating System

The existing hierarchy, people, and management still have a purpose and largely remain in place. However, SAFe creates a second *virtual* operating system organized around [Development Value Streams](#) instead of functional silos (or departments) to form the entrepreneurial network.

Each development value stream creates one or more [Agile Release Trains \(ARTs\)](#) with a shared business and technology mission. Each ART plans, commits, develops, and deploys together. They are an integral part of the entrepreneurial network that develops innovative products ([Solutions](#) and services).

Although the management reporting structure in the hierarchy may remain largely the same, the teams in an ART are self-organizing and self-managing, and they no longer need daily task direction. ARTs are virtual organizations with the people needed to define, deliver, and operate the solution. This new virtual organization breaks down the traditional functional silos that inhibit flow and innovation.

By organizing the second operating system around value streams instead of departments, SAFe offers a way for organizations to focus on customers, products, innovation, and growth in harmony with their existing hierarchical structure.

Moreover, this operating system is *flexible*. It is built on time-tested Lean-Agile SAFe practices. It can organize and *quickly reorganize* without completely disrupting the existing hierarchy, as illustrated in Figure 5. That's what business agility demands.

Figure 5. SAFe as the second organizational operating system

Responding to market changes and emerging opportunities is vital to surviving in the digital age, where disruption is the norm rather than the exception.

Technological advances have drastically shifted the dynamics of the competitive game by opening up various ways to win in the market. The ability to respond quickly with innovative business solutions—what we call Business Agility—is the deciding factor between success and failure. Therefore, enabling business agility is a mission-critical goal for every organizational leader.

An organization must first understand and then apply the SAFe® Business Agility Value Stream to deliver value faster to keep today’s customers and win new ones. The SAFe *Business Agility Value Stream (BAVS)* helps organizations visualize the steps and implement the SAFe core competencies needed to move from identifying an opportunity to delivering customer value in the shortest possible time. The following sections describe the BAVS in greater detail.

Core Competencies of Business Agility

Achieving business agility requires this dual operating system and a significant degree of expertise across SAFe’s seven core competencies, as illustrated in Figure 10. Fortunately, they are part of SAFe, and as organizations adopt the Framework, they acquire the competencies incrementally over time. While each competency can deliver value by itself, true business agility is only present when the enterprise achieves significant mastery of all. It’s a tall order, but the path is clear.

Figure 6. SAFe Core Competencies

Figure 6. SAFe Core Competencies

Each competency is described in an article of that same name. You can easily navigate each competency from the SAFe Overview tab or the SAFe home page.

Why the Business Agility Value Stream (BAVS)?

Digital transformation is advancing in almost all business processes. Technologies such as AI, big data, and cloud computing are unlocking possibilities for creating new customer value. New business opportunities emerge more frequently, and many have the potential to disrupt market incumbents. Companies that continuously leverage these technologies will acquire more customers and improve value to existing customers. Ultimately, they will dominate their markets. For example:

Customers who can seamlessly fulfill their primary banking needs with an easy-to-use mobile app will remain loyal to that bank.

Health organizations that pioneer virtual urgent care at home will earn the gratitude and loyalty of all those who need their help.

A driver whose vehicle gets smarter and safer every day is more likely to remain loyal to that automotive manufacturer.

Traditional Development Won't Get You There

In this new reality, competitiveness equates to rapidly delivering digitally enabled solutions. As Figure 7 illustrates, responding to a traditional, phase-gate development process may mean missing the opportunity.

Figure 7. The traditional approach to leveraging a business opportunity is just too slow

Waiting for a funding cycle, big-design-upfront, and a long development cycle results in delayed technical and customer feedback. And more than likely, the learning at the end of this cycle isn't positive. Simply put, it is difficult, if not impossible, to understand and adapt to customer needs and quickly deliver a solution with a traditional development approach.

Introducing the Business Agility Value Stream

Instead, what is needed is a rapid cycle of sensing and responding that helps the Enterprise navigate the unknowns and arrive at a desirable solution **before** the window of opportunity closes. This is the business agility value stream (BAVS), illustrated in Figure 2. And it is explicitly designed to foster rapid learning and enable more favorable business outcomes. By implementing SAFe, enterprises inherently develop the Lean, Agile, and DevOps capabilities that will allow incremental delivery at scale. Although these capabilities are essential, establishing true business agility requires flow to be cultivated and accelerated through the entire BAVS, from sensing an emerging opportunity to deliver the right solution (Figure 8).

Figure 8. A faster way to leverage a new business opportunity

Figure 8. A faster way to leverage a new business opportunity

Steps in the Business Agility Value Stream

While the specific implementation of the BAVS depends on the business context (organizational, market, customer, technology, and solution), the steps and basic

structure are essentially the same. In addition, the knowledge, skills, and behaviors required to succeed with each step of the BAVS are described in the related SAFe Core Competency, as we will see below.

This agility requires all functions, processes, activities, teams, and events from end to end to be aligned and optimized for maximum speed and quality. The following steps form this value stream:

Sense Opportunity

The first step, of course, is to be able to sense the opportunity!

This step requires the SAFe [Organizational Agility](#) core competency, which fosters activities such as:

- Market research
- Analysis of quantitative and qualitative data
- Direct and indirect customer feedback

Direct observation of the customers in the marketplace

Most directly savvy, Lean-thinking leaders Ÿgo see and spend significant time where customers perform work. They return with current, relevant, direct, and specific information about the realities of their products and services and identify opportunities for innovative new solutions.

Fund MVP

An enterprise must be able to respond quickly to these opportunities with nimble funding. [Lean Portfolio Management](#) (LPM) is the core competency needed to support this step.

With LPM, Lean Budgeting provides the ability to quickly allocate sufficient funds to build a Minimum Viable Product (MVP) —an early version of the solution used to evaluate the primary business hypothesis. The “Minimum” in MVP refers to the low cost of the experiment needed to test the hypothesis and

establish solution viability.

These funding decisions become visible and addressed as the new initiative flows through the SAFe Portfolio Kanban system.

Organize Around Value

The next step is to organize—nor reorganize as necessary—to address the new opportunity. An MVP can often be built by existing Agile Teams or Agile Release Trains (ARTs) as the new work find its way into their respective backlogs. However, creating an MVP may also involve modifying existing teams and ARTs. An entirely new development value stream may need to be formed in a more extreme case.

Two core competencies—[Team and Technical Agility](#) and [Organizational Agility](#)—enable this flexibility.

Connect to Customer

Agile development is inherently focused on assuring a direct connection to the customer, and Customer Centricity is the mindset that underpins it. This way of doing business focuses on creating positive experiences for the customer across the enterprise’s complete set of products and services and throughout the entire customer journey.

Design Thinking provides the tools that help teams and ARTs achieve these ideals by empathizing with the user to design the right solution. [Agile Product Delivery](#) is the core competency that enables this connection.

Deliver MVP

The proof is in the doing. Agile teams

and ARTs deliver the MVP iteratively and incrementally, following Lean-Agile practices.

However, there are differences in how teams and trains work on an MVP compared to evolving functionality in a mature solution. There is more risk and uncertainty for a start. Unknowns may manifest in critical areas, including technology choices, implementation strategy, organizational expertise, deployment, operations, customer acceptance, and business benefits. More experimentation and even faster feedback are required. But that is exactly what SAFe is optimized for.

Depending on the scope of the solution, Agile Product Delivery and [Enterprise Solution Delivery](#) are the two core competencies that enable this step.

Pivot or Persevere

The result of the MVP is a set of facts that support a decision regarding whether to proceed with further solution development. If the hypothesis is disproven, the organization accepts the sunk cost and moves on to other business opportunities. If the hypothesis proves beneficial, additional funding follows to enable further development. However, the MVP outcome is not always a simple yes or no. The experiment may yield vital insights that reveal new alternative solutions.

This decision point is a crucial investment milestone and is a critical stage in the portfolio kanban system. Again, Lean Portfolio Management is the core competency that enables this step.

Deliver Value Continuously

A successful MVP that confirms the hypothesis opens the gates to deliver value continuously with additional solution features. This process relies on Agile Product Delivery that fosters iterative and incremental development powered directly by the ART.

Building on DevOps, these practices include optimizing a Continuous Delivery Pipeline that ensures a steady

flow of value and the ability to release on-demand to meet the needs of customers and businesses.

For some organizations, these solutions represent large, significant, and complex applications and cyber-physical systems that require thousands of developers and many capable suppliers to coordinate their efforts within a Solution Train. Enterprise Solution Delivery is the core competency enabling this step in this case.

Learn and Adapt

Learn and Adapt is not the final step. A single initiative rarely determines business outcomes. Instead, the enterprise learns from the BAVS and the process and adapts based on these learnings.

Measurement is an integral part of improvement. As Figure 8 illustrates, three measurement domains—Competency, Flow, and Outcomes—

provide critical perspectives and measures of organizational performance that help identify impediments and opportunities for improvement.

The **Continuous Learning Culture** core competency is the primary driving force behind positive change. A learning organization has a sense of urgency, constantly looking for new business opportunities and improvements in existing processes and solutions. Indeed, the BAVS is an example of continuous learning, enabling continuous innovation. In addition, other forms of learning and adaptation happen through regular Inspect & Adapt events at every level of the SAFe operating system.

Lean-Agile Leadership Enables the BAVS

None of this happens without [Lean-Agile Leadership](#), the foundation of SAFe and the BAVS. The BAVS represents a new way of working for most enterprises, which is substantially different from the status quo and affects most aspects of a modern enterprise.

Lean-Agile leaders view the organization as a dynamic set of business agility value streams that pursue and leverage critical business opportunities. And importantly, they lead the change to arrive at this new state. After that, they focus the organization on successful BAVS execution and improving BAVS performance over time. Only then can business agility be achieved.

Measuring the Business Agility Value Stream

As Figure 9 illustrates, SAFe provides three measurement domains *flow*, *outcomes*, and *competency* suitable for measuring and improving any value stream, including the all-important BAVS.

Figure 8. Three SAFe measurement domains support the goal of business agility

Figure 9. Three SAFe measurement domains support the goal of business agility

Flow metrics help determine how fast the value stream creates and delivers value, represented by flow *distribution, velocity, time, load, efficiency, and predictability*.

Outcomes metrics help ensure the solution delivered benefits the customer and the business. [Value stream KPIs](#) are primarily used to measure these outcomes.

Competency measures evaluate organizational proficiency on two levels:

1. The [SAFe Business Agility Assessment](#) offers business and portfolio stakeholders a way to assess their overall progress.
2. The individual [SAFe Core Competency Assessments](#) help teams and ARTs improve their technical and business practices to achieve the portfolio's goal.

See the [Measure and Grow](#) article for more details.

The road to business agility is long and never-ending. Measuring it helps enterprises understand where they are on their journey and reminds them to celebrate small successes.

Summary

Welcome to the age of software and digital, where business agility will be the most significant factor in deciding the winners and losers in the new economy:

Lean-Agile *commercial* businesses will create higher profits, increase employee engagement, and more thoroughly satisfy customer needs

Lean-Agile *nonprofits* will build resilience, sustainability, and the alignment needed to fulfill their mission

Lean-Agile *governments* will deliver systems that better ensure the public's safety, economy, and general welfare

These three market segments depend on delivering innovative business solutions faster and more efficiently than ever. Each will need a dual operating system: a hierarchy for efficiency and scale and a second, customer-centric network operating system that delivers innovative solutions. SAFe's seven core competencies will enable this all-important dual operating system. Those who master these competencies will survive and thrive in the digital age.

Learn More

[1] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. IT Revolution Press, 2018.

[2] Perez, Carlota. *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages*. Edward Elgar Publishing, 2002.

[3] Kotter, John. *XLR8 (Accelerate): Building Strategic Agility for a Faster-Moving World*. Harvard Business Review Press, 2014.

[4] [https://www.investopedia.com/articles/investing/072115/companies-went-bankrupt-innovation lag](https://www.investopedia.com/articles/investing/072115/companies-went-bankrupt-innovation-lag)

Last update: 13 February 2023

images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Real learning gets to the heart of what it means to be human. Through learning, we recreate ourselves. Through learning, we become able to do something we never were able to do. Through learning, we re-perceive the world and our relationship to it. Through learning, we extend

Adjust Cookie Settings

*our capacity to
create, to be part of
the generative
process of life.*

*There is within each
of us a deep hunger
for this type of
learning.*

ÑPeter M. Senge,
The Fifth Discipline

Continuous Learning Culture

This is achieved by becoming a learning organization, committing to relentless improvement, and promoting a culture of innovation.

It is one of the seven core competencies of Business Agility, each of which is essential to achieving Business Agility. Each core competency is supported by a specific assessment, enabling the enterprise to assess its proficiency. The Measure and Grow article provides these core competency assessments and recommended improvement opportunities.

Why Continuous Learning Culture?

Organizations today face an onslaught of forces that create both uncertainty and opportunity. The pace of technological innovation is beyond exponential. Startup companies challenge the status quo by transforming, disrupting, and in some cases eliminating entire markets. Juggernaut companies like Amazon and Google are entering new markets like banking and healthcare. At any moment, political, economic, and environmental turmoil threatens to change the rules. Expectations from new generations of workers, customers, and society challenge companies to think and act beyond balance sheets and quarterly earnings reports. Due to these factors and more, one thing is sure: organizations in the digital age must be able to adapt rapidly and continuously or face declineÑand, ultimately, extinction.

WhatÕs the solution? Organizations must evolve into adaptive engines of change to thrive in the current climate, powered by a fast and effective learning culture. Learning organizations leverage the collective knowledge, experience, and creativity of their workforce, customers, supply chain, and the broader ecosystem. They

harness the forces of change to their advantage. In these enterprises, curiosity, exploration, invention, entrepreneurship, and informed risk-taking replace commitment to the status quo while providing stability and predictability. Rigid, siloed top-down structures give way to fluid organizational constructs that can shift as needed to optimize the flow of value. Decentralized decision-making becomes the norm as leaders focus on vision and strategy and enable organization members to achieve their fullest potential.

Any organization can begin the journey to a continuous learning culture by focusing its transformation on three critical dimensions, as shown in Figure 1.

Figure 1. The three dimensions of a continuous learning culture

Figure 1. The three dimensions of a continuous learning culture

The three dimensions are:

1. **Learning Organization** Employees at every level are learning and growing so that the organization can transform and adapt to an ever-changing world.

2. **Innovation Culture** Ⓛ Employees are encouraged and empowered to explore and implement creative ideas that enable future value delivery.
3. **Relentless Improvement** Ⓛ Every part of the enterprise focuses on continuously improving its solutions, products, and processes.

The sections below describe each of these dimensions.

Learning Organization

Learning organizations invest in and facilitate the ongoing growth of their employees. When everyone in the organization continuously learns, it fuels the enterprise's ability to dynamically transform itself to anticipate and exploit opportunities that create a competitive advantage. Learning organizations excel at creating, acquiring, and transferring knowledge while modifying practices to integrate new insights [1,2]. These organizations understand and foster people's intrinsic nature to learn and gain mastery, harnessing that impulse for the benefit of the enterprise [3].

Learning organizations are different from those using the scientific management methods promoted by Frederick Taylor. In Taylor's model, learning is limited to those at the top while everyone else follows the policies and practices created by management. Becoming a learning organization is not an altruistic exercise. It's an antidote to the status-quo thinking that drove many former market leaders to bankruptcy. Learning drives innovation, leads to greater information sharing, enhances problem-solving, increases the sense of community, and surfaces opportunities for more efficiency. [4]

The transformation into a learning organization requires five distinct disciplines, as described by Senge. The best practices for developing these disciplines include:

Personal Mastery Ⓛ Employees develop as "T-shaped" people who build a breadth of knowledge in multiple disciplines for efficient collaboration and deep expertise aligned with their interests and skills. T-shaped employees are a critical foundation of Agile teams.

Shared Vision Ⓛ Forward-looking leaders envision, align with, and articulate exciting possibilities. Then, they invite others to share and contribute to a common view of the future. The vision is compelling and motivates employees to contribute to achieving it.

Team Learning Ⓛ Teams work collectively to achieve common objectives by sharing knowledge, suspending assumptions, and "thinking together." They complement each other's skills for group problem-solving and learning.

Mental Models □ Teams surface their existing assumptions and generalizations while working with an open mind to create new models based on a shared understanding of the Lean-Agile way of working and their customer domains. These models make complex concepts easy to understand and apply.

Systems Thinking □ The organization sees the larger picture and recognizes that optimizing individual components does not optimize the system. Instead, the business takes a holistic learning, problem-solving, and solution-development approach. This optimization extends to business practices such as Lean Portfolio Management (LPM), which ensures that the enterprise invests in experimentation and learning to drive the system forward.

Many of SAFe's principles and practices directly support these efforts, as illustrated in Figure 2.

Figure 2. SAFe includes principles and practices that support the learning organization.

Here are some of the ways SAFe promotes a learning organization:

Lean-Agile leaders who are insatiable learners use successes and failures in SAFe practices as learning moments to build mastery.

A shared vision is iteratively refined during each PI Planning period. This shared vision influences Business Owners, the teams on each Agile Release Train (ART), and the entire organization.

Teams learn continuously through daily collaboration and problem-solving, supported by events such as team retrospectives and Inspect & Adapt.

Systems Thinking is a cornerstone of Lean-Agile and one of the ten SAFe principles. SAFe also provides regular dedicated time and space for learning through the Innovation and Planning (IP) iteration that occurs every PI. People working in a SAFe organization are encouraged to build learning networks across organizational boundaries and outside the organization. (Learning networks consist of trusted connections with whom an individual interacts and learns from regularly.)

Innovation Culture

An organization's innovativeness is essential to competing in the digital age. Such efforts cannot be infrequent or random. It requires an *innovation culture*. An innovation culture exists when leaders create an environment that supports creative thinking and curiosity and challenges the status quo. When an organization has an innovation culture, employees are encouraged and enabled to:

- Explore ideas for enhancements to existing products
- Experiment with ideas for new products
- Pursue fixes to chronic defects
- Create improvements to processes that reduce waste
- Remove impediments to productivity

Some organizations support innovation with paid time for exploring and experimenting, intrapreneurship programs, and innovation labs. SAFe goes further by providing consistent time each PI for all Agile Release Train (ART) participants to pursue innovation activities during the Innovation and Planning (IP) iteration. Innovation is also integral to Agile Product Delivery and the Continuous Delivery Pipeline.

The following sections provide practical guidance for initiating and continuously improving an innovation culture.

Innovative People

The foundation of an innovation culture recognizes that systems and cultures don't innovate: people innovate. Instilling innovation as a core organizational capability requires cultivating the courage and aptitude for innovation and encouraging employee risk-taking. For existing organization members, this may necessitate coaching, mentoring, and formal training in the skills and behaviors of entrepreneurship and innovation. Individual goals and learning plans should include language that enables and empowers growth as an innovator. Rewards and recognition that balance intrinsic and extrinsic motivation reinforce the importance of everyone as an innovator. Criteria for hiring new employees should include

evaluating how candidates will fit in an innovation culture. Opportunities and paths for advancement should be clear and available for people who demonstrate exceptional talent and performance as innovation agents and champions [5].

Time and Space for Innovation

Building time and space for innovation includes providing work areas conducive to creative activities and setting aside dedicated time from routine work to explore and experiment. Innovation space can also include:

Broad cross-domain interactions involving customers, the supply chain, and even the physical or professional communities connected to the organization

Temporary and limited suspension of norms, policies, and systems (within legal, ethical, and safety boundaries) to challenge existing assumptions and explore what's possible

Systematic activities (IP iteration, hackathons, dojos, and so on) and opportunistic innovation activities (continuous, accidental, unplanned)

Perpetual innovation forums on collaboration platforms and Communities of Practice (CoPs) create the opportunity for ongoing conversations across the organization.

Go See

The best innovation ideas are often sparked by seeing the problems to be solved first-hand—witnessing how customers interact with products or the challenges they face using existing processes and systems. Gemba is a Lean term and practice from Japan, meaning “the real place,” where the customers’ work is performed. SAFe explicitly supports this concept through Continuous Exploration. First-hand observations and hypotheses channel the creative energy of the entire organization toward conceiving innovative solutions. Leaders should also openly share their views on the opportunities and challenges the organization faces to focus innovation efforts on the things with the highest potential to benefit the enterprise.

Experimentation and Feedback

Innovation cultures embrace the idea that conducting experiments designed to progress iteratively towards a goal is the most effective path to learning that creates successful breakthroughs. Regarding the many unsuccessful experiments to make an incandescent light bulb, Thomas Edison famously said, “I have not failed. I’ve just found 10,000 ways that won’t work.” Experiments don’t fail in the scientific method; they produce the data needed to accept or reject a hypothesis. Many companies don’t innovate sufficiently due to a fear of failure culture. Such fear cripples innovation.

In contrast, innovation cultures depend on learning from experiments and incorporating those insights into future exploration. When leaders create the psychological safety described in the Lean-Agile Leadership article, people are encouraged to experiment (within guardrails). They feel they have permission to solve big problems, seize opportunities, and do so without fear of blame, even when the results of the experiments suggest moving in a different direction.

Pivot Without Mercy or Guilt

Every innovation begins as a hypothesis – a set of assumptions and beliefs regarding how a new or improved product will delight customers and help the organization achieve its business objectives. However, hypotheses are just informed guesses until real customers provide validated feedback. As Eric Ries promotes in *The Startup Way*, the fastest way to accept or reject a product development hypothesis is to experiment by building a Minimum Viable Product or MVP [6]. An MVP is the simplest thing that can possibly work to test the proposed innovation to see if it leads to the desired results. Customers and intended users of the system must test MVPs in the target market for fast feedback. In many cases, the feedback is positive, and further investment is warranted to bring the innovation to market or into production. In other instances, the feedback dictates a change in direction. This change could be as simple as a set of modifications to the product followed by additional experiments for feedback, or it could prompt a “pivot” to an entirely different product or strategy. When the fact-based evidence indicates that a pivot is required, the shift in direction should occur as quickly as possible without blame and consideration of sunk costs in the initial experiments.

Innovation Riptides

Organizations must go beyond catchy slogans, “innovation teams,” and popular techniques like hackathons and dojos to create an innovation culture. A fundamental rewiring of the enterprise’s DNA is needed to fully leverage the innovation mindset and develop the processes and systems that promote sustained innovation. As shown in Figure 3, SAFe provides these required structures.

Figure 3. SAFe includes critical elements to support a consistent, continuous flow of innovation

The continuous flow of innovation is built on SAFe principle #9, which promotes decentralized decision-making. Some innovation starts as strategic portfolio concerns realized through Epics and Lean Budgets applied to value streams. In building the solution to realize Epics, teams, suppliers, customers, and business leaders identify opportunities for improving the solution. The potential innovations that result can be considered an “innovation riptide” that flows back into the structures SAFe provides for building solutions. Smaller, less expensive innovations flow into the ART Kanban as Features. In contrast, more significant, costly innovations require an Epic and Lean Business Case and flow into the Portfolio Kanban.

Relentless Improvement

Since its inception in the Toyota Production System, *kaizen*, or the relentless pursuit of perfection, has been one of the core tenets of Lean. While unattainable, striving for perfection leads to continuous improvements to products and services. In the process, companies have created more and better products for less money and with happier customers, leading to higher revenues and greater profitability. Taiichi Ohno, the creator of Lean, emphasized that the only way to achieve kaizen is for every employee always to have a mindset of continuous improvement. The entire enterprise as a system—executives, product development, accounting, finance, and sales—is continuously being challenged to improve [7].

But improvement requires learning. Rarely are the causes and solutions for problems that organizations face clear and easily identified. The Lean model for

continuous improvement is based on small iterative and incremental improvements and experiments that enable the organization to learn its way to the most promising answer to a problem.

Relentless improvement is one of the four SAFe Core Values, conveying that improvement activities are essential to the survival of an organization and should be given priority, visibility, and resources. The following sections illustrate how a continuous learning culture is a critical component of relentless improvement.

Constant Sense of Urgency

Succeeding in the digital age requires sensing shifting market conditions and responding quickly. It requires inviting continuous feedback from customers even if the learning gained leads to change. Delivering needed improvements rapidly is as important as identifying what needs to change. Faster time-to-market requires a bias for action and a constant sense of urgency. In SAFe, this means addressing time-critical improvements frequently. Agile teams make improvements daily as needed and through the effective use of cadence-based SAFe events such as team retrospectives, the problem-solving workshop during Inspect & Adapt (I&A), and the IP iteration. Improvement Features and Stories that emerge from the I&A are incorporated into team plans and prioritized in work planned for the following PI. Time-critical improvements are addressed even more quickly using techniques such as an expedite lane in team and ART Kanbans or simply pausing routine work to swarm on high-impact issues.

Problem Solving Culture

In Lean, problem-solving is the driver for continuous improvement. It recognizes that a gap exists between the current and desired states, requiring an iterative process to achieve the target state. The steps of problem-solving are both fractal and scalable. They apply to teams trying to optimize response time in a software system and to enterprises attempting to reverse a steady decline in market share. Iterative Plan-Do-Check-Adjust (PDCA) cycles, as shown in Figure 4, provide the process for iterative problem solving that is applied until the target state is achieved. This model treats problems as opportunities for improvement in a blameless process. Employees at all levels are empowered and equipped with the time and resources to identify and solve problems. More importantly, every employee views solving problems as part of their ongoing responsibilities, empowered by decentralized decision-making (SAFe Principle #9). Techniques such as retrospectives, problem-solving workshops, hackathons, and communities of practice are ways SAFe reinforces a problem-solving culture.

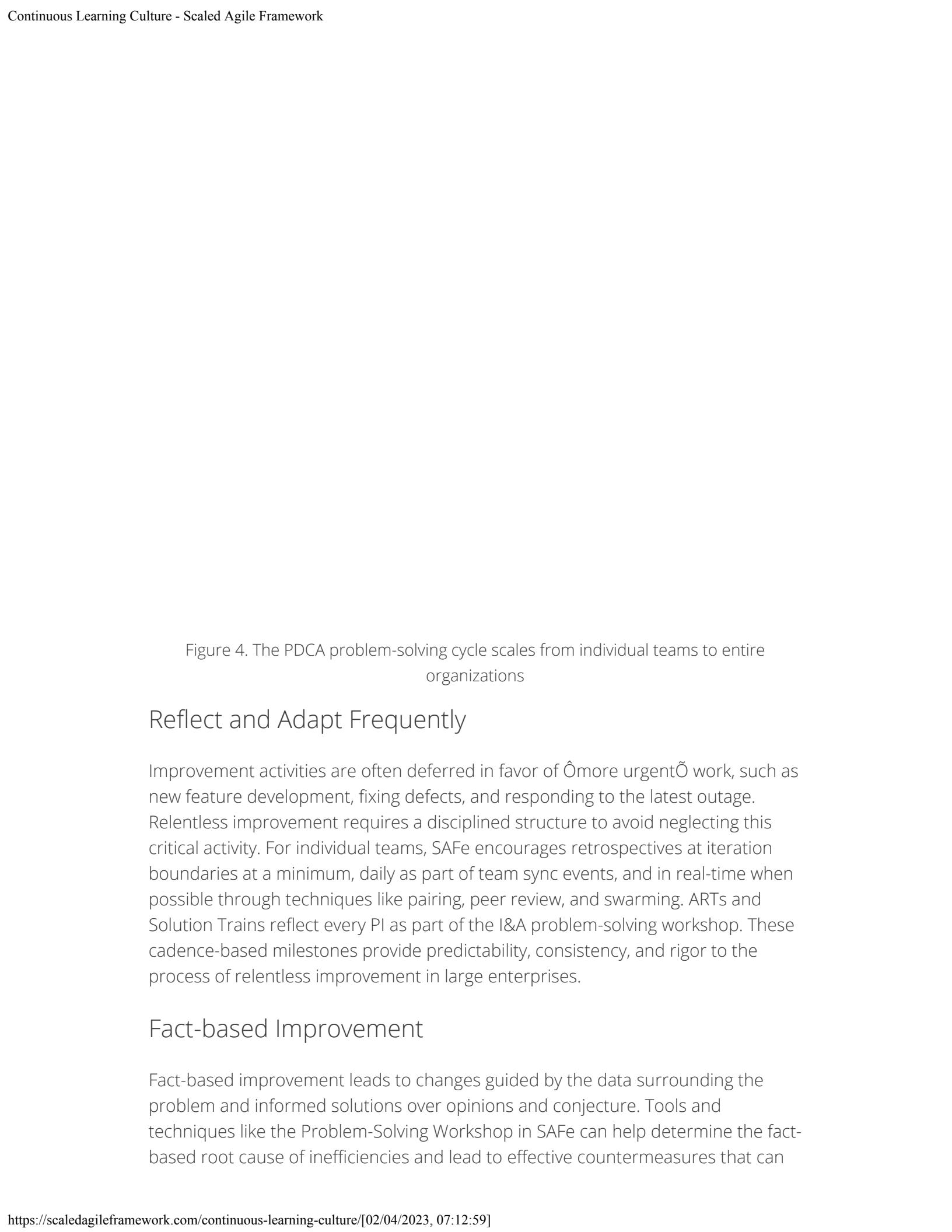


Figure 4. The PDCA problem-solving cycle scales from individual teams to entire organizations

Reflect and Adapt Frequently

Improvement activities are often deferred in favor of Ômore urgentÕ work, such as new feature development, fixing defects, and responding to the latest outage. Relentless improvement requires a disciplined structure to avoid neglecting this critical activity. For individual teams, SAFe encourages retrospectives at iteration boundaries at a minimum, daily as part of team sync events, and in real-time when possible through techniques like pairing, peer review, and swarming. ARTs and Solution Trains reflect every PI as part of the I&A problem-solving workshop. These cadence-based milestones provide predictability, consistency, and rigor to the process of relentless improvement in large enterprises.

Fact-based Improvement

Fact-based improvement leads to changes guided by the data surrounding the problem and informed solutions over opinions and conjecture. Tools and techniques like the Problem-Solving Workshop in SAFe can help determine the fact-based root cause of inefficiencies and lead to effective countermeasures that can

be applied rapidly. Root cause analysis is exponentially more effective when supported by data. The self-assessments described in Measure and Grow provide one type of data-driven feedback to help focus improvement work. Organizations practicing the disciplines outlined in Big Data have even more powerful tools and analytics to deliver data-driven insights that lead to more targeted and effective improvement efforts.

Optimize the Whole

“Optimize the whole” suggests that improvements should be designed to increase the effectiveness of the entire system that produces the sustainable flow of value instead of optimizing individual teams, silos, or subsystems. Organizing around value in ARTs, Solution Trains, and value streams creates opportunities for people in all domains to have regular cross-functional conversations about enhancing overall quality, the flow of value, and customer satisfaction. Participants in Lean Portfolio Management bring leaders together from across the organization to prioritize investments for improvements and new solutions holistically, representing a fundamental shift from past funding practices and prioritizing initiatives within silos.

Summary

Too often, organizations assume that the culture, processes, and products that led to today’s success will also guarantee future results. That mindset increases the risk of decline and failure. The enterprises that will dominate their markets in the future will be adaptive learning organizations with the ability to learn, innovate, and relentlessly improve more effectively and faster than their competition.

Competing in the digital age requires investment in time and resources for innovation, built upon a culture of creative thinking and curiosity—an environment where norms can be challenged, and new products and processes emerge. Alongside this, relentless improvement acknowledges that the survival of an organization is never guaranteed. Everyone in the organization will be challenged to find and make incremental improvements, and leaders will give priority and visibility to this work.

A continuous learning culture will likely be the most effective way for this next generation of workers to improve relentlessly and the successful companies that employ them.

Learn More

- [1] Garvin, David A. *Building a Learning Organization*. Harvard Business Review, July-August 1993. <https://hbr.org/1993/07/building-a-learning-organization>
- [2] Senge, Peter. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Penguin Random House LLC, 2006.
- [3] Pink, Daniel H. *Drive: The Surprising Truth About What Motivates Us*. Penguin Group, 2011.
- [4] Marquardt, Michael. *Building the Learning Organization*. Nicholas Brealey Publishing, 2011.
- [5] Beswick, Cris, Derek Bishop, and Jo Geraghty. *Building a Culture of Innovation*. KoganPage Publishing, 2015.
- [6] Ries, Eric. *The Startup Way*. Currency, 2017.
- [7] Liker, Jeffery K. *Developing Lean Leaders at All Levels*. Lean Leadership Institute Publications, 2014.

Last update: 23 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite

[presentations](#)

GET SOCIAL

[Permissions Form](#)

300

[Blog](#)

[Twitter](#)

[Usage and Permissions](#)

Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[Linkedin](#)

BUSINESS

[About Certification](#)

[YouTube](#)

HOURS

[Become a Trainer](#)

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*Agility is the ability
to adapt and
respond to change. Agile organizations
view change as an
opportunity, not a
threat.*

Jim Highsmith

Organizational Agility

The Organizational Agility (OA) competency describes how Lean-thinking people and Agile teams across the enterprise optimize their business processes, evolve strategy with clear and decisive new commitments, and quickly adapt the

Adjust Cookie Settings

organization as needed to capitalize on new opportunities.

Organizational Agility is one of the seven core competencies of Business Agility, each of which is essential to achieving Business Agility. Each core competency is supported by a specific assessment, which enables the enterprise to assess its proficiency. The Measure and Grow article presents these core competency assessments and recommended improvement opportunities.

Why Organizational Agility?

In today's digital economy, the only truly sustainable competitive advantage is the speed at which an organization can sense and respond to the needs of its customers. Its strength is its ability to deliver value in the shortest sustainable lead time, to evolve and implement new strategies quickly, and to reorganize to address emerging opportunities better.

Organizational agility is critical to respond sufficiently to challenges. Unfortunately, most businesses' organizational structures, processes, and cultures were developed more than a century ago. They were built for control and stability, not innovation, speed, and agility. Small incremental changes to how businesses manage, strategize, and execute are insufficient to remain competitive. This requires a leaner and more Agile approach which, in turn, requires sweeping changes that have a positive, long-lasting impact on the entire enterprise.

The SAFe approach to addressing the challenge of digital transformation is the "dual operating system" (see [Business Agility](#)). This approach leverages the stability and resources of the existing organizational hierarchy while implementing a value stream network that leverages the entrepreneurial drive still present in every organization. By organizing and reorganizing the enterprise around the flow of value instead of the traditional organizational silos, SAFe restores the second (network) operating system. It allows organizations to focus on the innovation and growth of new ideas and the execution, delivery, operation, and support of existing solutions.

The organizational agility competency is instrumental in bringing the power of the second operating system to support the opportunities and threats of the digital age. This competency is expressed in three dimensions (Figure 1):

Figure 1. Three dimensions of organizational agility

Figure 1. Three dimensions of organizational agility

Lean-Thinking People and Agile Teams ☐ Everyone involved in solution delivery is trained in Lean and Agile methods and embraces their values, principles, and practices.

Lean Business Operations ☐ Teams apply Lean principles to understand, map, and continuously improve the processes that deliver and support business solutions.

Strategy Agility ☐ The enterprise is Agile enough to sense the market and quickly change strategy when necessary.

Each of these dimensions is described in the sections that follow.

Lean-Thinking People and Agile Teams

Lean-thinking people and Agile teams is the first dimension of organizational agility. This dimension is critical to delivering business solutions—not just the

software applications and digital systems, but all supporting activities (privacy, security, support, availability) necessary to continually address the business problem. Even though the solution is not stand-alone, it lives in the larger context of its environment—*including other hardware, software, network systems, and more.*

Everyone involved in delivering business solutions—operations, legal, marketing, people operations, finance, development, and others—can apply effective Lean and Agile methods and embrace the mindset, values, principles, and practices.

Extending the Mindset, Values, and Principles to the Enterprise

Extending the [Lean-Agile mindset](#) to the entire enterprise forms the cornerstone of a new management approach and results in an enhanced company culture that enables business agility. It provides leaders and practitioners throughout the enterprise with the thinking tools and behaviors needed to drive a successful SAFe transformation, helping individuals and the entire enterprise achieve their goals.

The Lean-Agile mindset establishes correct thinking, but it’s the ten underlying [SAFe principles](#) that guide effective roles, practices, and behaviors. These principles are fundamental to organizational agility and the Lean-thinking people and Agile teams who enable it. Everyone in the enterprise can apply these principles to their daily work and become part of the leaner and more Agile operating system.

As Agile Teams begin to form, we’ve often observed a “three-step maturity cycle” which illustrates how they develop and mature the practices that are particular to their domain. (Figure 2):

Figure 2. The Agile team maturity cycle

Figure 2. The Agile team maturity cycle

Step 1: Be Agile

First, the teams adopt and master the Lean-Agile mindset and practices. This creates a universal value system and a shared understanding of Agile. SAFe's Lean-Agile principles and Lean-Agile Mindset guide the right thinking and behaviors for teams and their leaders. They provide a "North Star" that points the way to being Agile, even when specific Agile guidance does not exist for that domain.

Step 2: Know your Value Stream

Next, teams must know how they participate in the organization's flow of value within the operational and development value streams. Taking a system's view of the value delivery process allows teams to understand where and with who most of their interactions occur. [Value Stream Mapping](#) helps understand the steps in value delivery and the boundaries of the value stream. Based on this knowledge, teams organize to increase productivity and deliver business value.

Step 3: Specialize the Principles and Practices

As teams and individuals mature, they must evolve their practices to define what Agile and built-in quality mean in their context. In this way, they make it their own. While driven by the same set of principles, the practices, and the methods in which they are applied, differ. Organizing around value, facilitating flow, planning, synchronizing, reviewing results, building quality, and delivering value depend on a set of unique parameters within which each team operates. Teams pick a simple starting point and continually adjust their way of working based on their individual experience with a set of practices. As already discussed here, many business and technology domains have already begun this journey.

Applying SAFe to Business *and* Technology Teams

As described in the [Team and Technical Agility](#) competency article, the emergence of Agile software development is fairly advanced and well understood. With the advent of the DevSecOps movement, IT operations and security are also rapidly adopting Agile. Agile is also making its way to other technical domains, such as networking, operations (see [DevOps](#)), hardware (see [Applying SAFe to Hardware Development](#)), electronics, and more. After all, Agile technical teams typically achieve a degree of unprecedented performance and personal satisfaction with their work. Who doesn't want to be on a high-performing Agile team?

In addition, once the business understands this new way of working, it begins recognizing that the same benefits apply and typically starts creating cross-functional Agile business teams. These teams may be involved in any of the functions necessary to support developing and delivering business solutions, including the following:

- Sales, product marketing, and corporate marketing (see [SAFe for Marketing](#))
- Sourcing and supply chain management
- Operations, legal, contracts, finance, and compliance (see [Agile Contracts](#))
- People operations (see [Agile HR](#))
- Receiving, production, fulfillment, and shipping
- Customer service, support, and maintenance

Additionally, Built-in Quality practices in SAFe apply to Agile Business Teams, helping them improve the value of deliverables and achieve faster flow. No matter your business function, the steps to achieve quality with Agility include the following:

- Organize into Agile teams, get trained, and iterate
- Define the standards and compliance policies for your function
- Agree on the definition of done (DoD) for artifacts and activities for your workflow

- Implement the basic Agile quality practices
- Measure and learn. Specialize Agile quality practices further to your specific function
- Improve relentlessly

Patterns for Agile Business and Technology

While Agile Teams are essential to a SAFe enterprise, they are not enough. Enabling Organizational Agility requires higher-level constructs that facilitate responsiveness and speed enterprise operations. In addition to the foundational SAFe concepts such as ART, Solution Train, and Portfolio, the following organizational patterns have proven effective in extending agility throughout the enterprise:

Pattern	Key characteristic	Importance to Organizational Agility
Business-Enabled ART	Includes the business people needed to deliver a full business solution	Integrates the business knowledge required to make effective solutions
Agile Business Train	Builds and operates a full business solution	Empowers employees and teams to define, build, test, deploy, support, and commercialize innovative business solutions with the shortest possible lead times
Agile Executive Team	Exemplar, cross-functional Agile team with Team Coach and PO	Speaks with one voice, fosters a more productive decision-making process, and instills a Lean-Agile Mindset by leading by example
Agile Business Function	Applies Lean-Agile practices to streamline business operations	Improves speed and quality of the business function, improves interactions with organizational value streams, and provides transparency of workflow
Combined Portfolio	Includes operational and development value streams	Enhances strategy and governance with a holistic view of operations and development of solutions within a Portfolio

Additional guidance on these important structures appears in the [Business and Technology](#) article.

Agile Working Environments

To fully benefit from the Agile operating model, organizations design work environments that support collaboration, transparency, and the ability to focus on a task at hand. This is especially important in today's highly distributed organizations. The [Agile Workspaces](#) article provides in-depth guidance on how to organize the high-performing Agile environment effectively.

Lean Business Operations

Lean business operations is the second dimension of organizational agility. Organizational agility requires enterprises to understand both the [Operational Value Streams](#) (OVS) that deliver business solutions to their customers and the [Development Value Streams](#) (DVS, which are the primary focus of SAFe) that develop those solutions.

Figure 3 illustrates the relationship between operational and development value streams.

Figure 3. Operational and development value streams

Figure 3. Operational and development value streams

For many developers, the people who run the operational value streams are the customers of the development value streams. They directly use, operate, and support the solutions that support the flow of value to the end user. This requires that developers:

Understand (and often help analyze and map) the operational value streams they support

Apply customer-centricity and design thinking to internal **and** external solutions
Include the business teams that support the solution in the development process

These responsibilities help assure that the business solutions developed provide a Ôwhole-product solutionÕ to satisfy the needs of both internal and external customers.

Mapping Value Streams

Lean companies focus on value streams to eliminate non-value-creating

*activities. Good development systems consistently create profitable (operational) value streams.*Ó

ÑAlan Ward

Identifying operational and development value streams is critical for every Lean enterprise. Once identified, value stream mapping is used to analyze and improve business operations [4]. Figure 4 illustrates a simplified example of value stream mapping, which, in this case, shows a few of the steps in a marketing program launch.

Figure 4. Value stream mapping showing the total lead time, total processing time, and time efficiency

Teams look for the opportunity to improve the efficiency of each step, consequently reducing the total lead time. This includes reducing process time and improving the quality of each step measured by the percentage complete and

accurate. (The percent complete and accurate represents the percentage of work that the next step can process without it needing to go back for rework.)

As is the case in the figure above, the delay time (the waiting time between steps) is often the most significant source of waste. If the team above wanted to deliver a marketing campaign faster, they would need to reduce the delay times, as the processing steps are only a small part of the total lead time. Reducing delays is typically the fastest and easiest way to shorten the total lead time and improve time to market.

Implementing Flow

SAFe Principle #6, [Make Value Flow Without Interruptions](#), provides good guidance for establishing fast flow through the system. It includes accelerators that can use to improve flow in every organizational environment. For example, every flow system has a bottleneck constraining its throughput (Figure 5). Addressing the current bottleneck is one of the flow accelerators. Applying it elevates flow performance to a new level and reveals new bottlenecks that need to be addressed.

Figure 5. Bottlenecks reduce the flow of value through the value stream

Figure 5. Bottlenecks reduce the flow of value through the value stream

Implementing flow is a comprehensive task that involves this and other accelerators that span from workers and their work environment to process steps and work items that pass through them.

Strategy Agility

Strategy agility is the third dimension of organizational agility. Strategy agility is the ability to sense changes in market conditions and implement new strategies quickly and decisively when necessary. It also includes the good sense to persevere on the things that are working—nor will work—if given sufficient time and focus. Figure 6 illustrates how the strategy must respond to market dynamics to successfully realize the enterprise’s mission.

Figure 6. Strategy responds to market dynamics

Figure 6. Strategy responds to market dynamics

Enterprises that have mastered strategy agility typically exhibit several capabilities, including those described in the following sections.

Market Sensing

Market sensing represents the culture and practice of understanding changing market dynamics based on the following:

- Market research
- Analysis of quantitative and qualitative data
- Direct and indirect customer feedback
- Direct observation of the customers in the marketplace

Savvy, Lean-thinking leaders often see and spend significant time in the place where the customer's work is actually performed. They return with current, relevant, and specific information about the realities of their products and services instead of opinions filtered through other perspectives.

Innovating Like a Lean Startup

After sensing opportunities, the Lean enterprise visualizes and manages the flow of new initiatives and investments by adopting a “build-measure-learn” Lean startup cycle. These initiatives are often new business solutions but may also be new business processes and capabilities that use existing solutions. Testing the outcome hypothesis via a Minimum Viable Product (MVP) before committing to a more significant investment reduces risk while generating helpful feedback.

Implementing Changes in Strategy

Identifying and defining a new strategy is only the first step. Once determined, the strategy must be communicated to all stakeholders in a new vision and roadmap and then, of course, be implemented. After all, significant strategy changes often affect multiple portfolio solutions and require coordination and alignment. Consequently, most large strategy changes require new epics to implement the change across value streams.

Figure 7 illustrates how new epics go through the various Kanban systems and backlogs that manage the flow of work. During the ordinary course of work, all backlogs are continuously reprioritized. Kanban systems help changes in strategy move quickly across value streams to the teams who do the implementation. This way, execution is aligned—and constantly realigned—to the evolving business strategy.

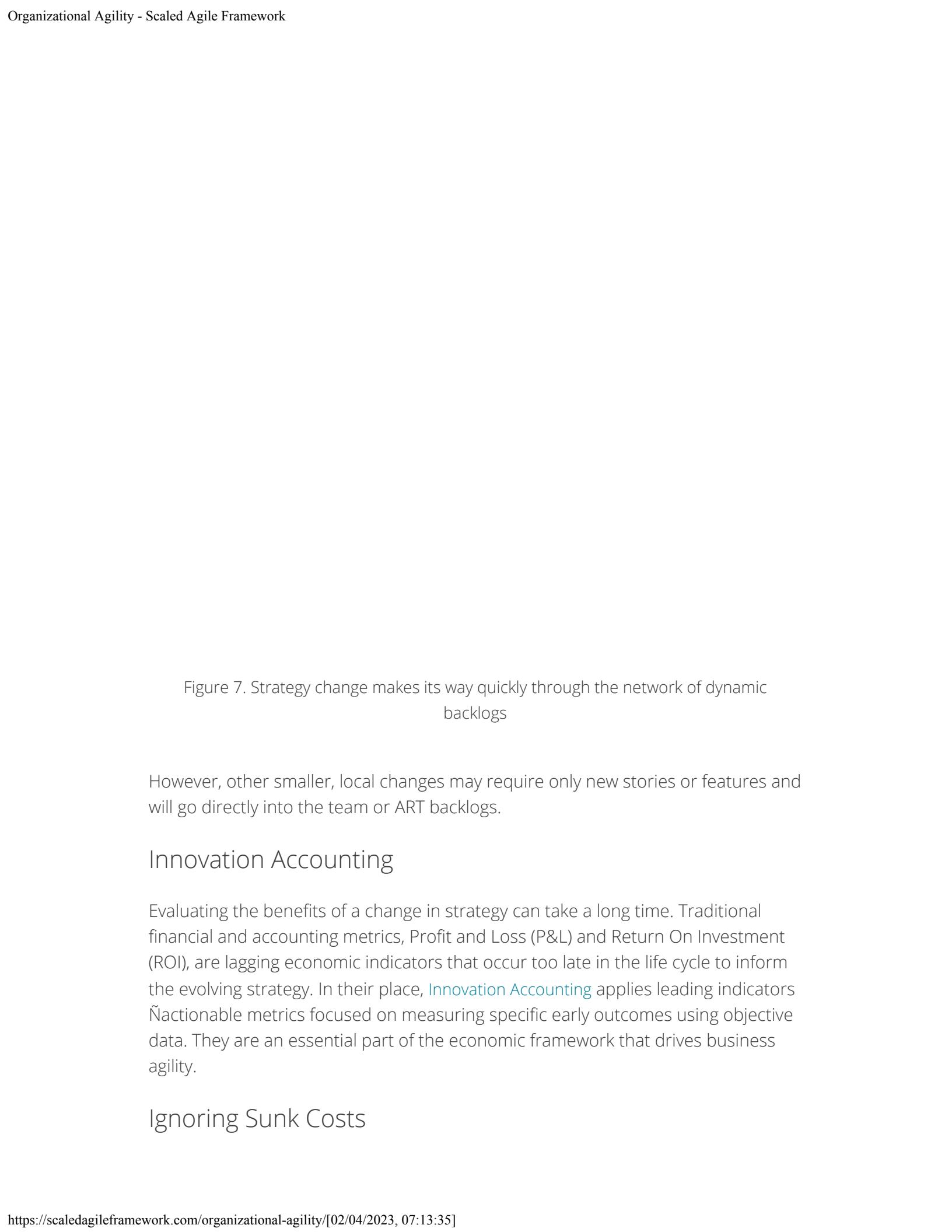


Figure 7. Strategy change makes its way quickly through the network of dynamic backlogs

However, other smaller, local changes may require only new stories or features and will go directly into the team or ART backlogs.

Innovation Accounting

Evaluating the benefits of a change in strategy can take a long time. Traditional financial and accounting metrics, Profit and Loss (P&L) and Return On Investment (ROI), are lagging economic indicators that occur too late in the life cycle to inform the evolving strategy. In their place, [Innovation Accounting](#) applies leading indicators —actionable metrics focused on measuring specific early outcomes using objective data. They are an essential part of the economic framework that drives business agility.

Ignoring Sunk Costs

A key factor in strategy agility is ignoring sunk costs, the expenses that have already occurred during solution development. Sunk costs cannot be recovered or changed and are independent of any future costs a company may incur [5]. Because strategic decision-making affects only the future course of business, sunk costs are absolutely irrelevant when evaluating a change in strategy. Instead, decision-makers should base all strategy decisions solely on the future costs of the initiatives necessary to achieve the change. When stakeholders do not have to waste energy to defend past spending, the organization can pivot more quickly to a new strategy.

Organizing and Reorganizing Around Value

Finally, [SAFe Principle #10 - Organize around value](#) guides enterprises to align their development efforts around the full, end-to-end flow of value. This principle highlights the dual operating system, one that leverages the benefits of the existing hierarchy but also creates a value stream network (Figure 8 below). This network assembles the people who need to work together, aligns them to the needs of the business and customer, minimizes delays and handoffs, and increases quality.

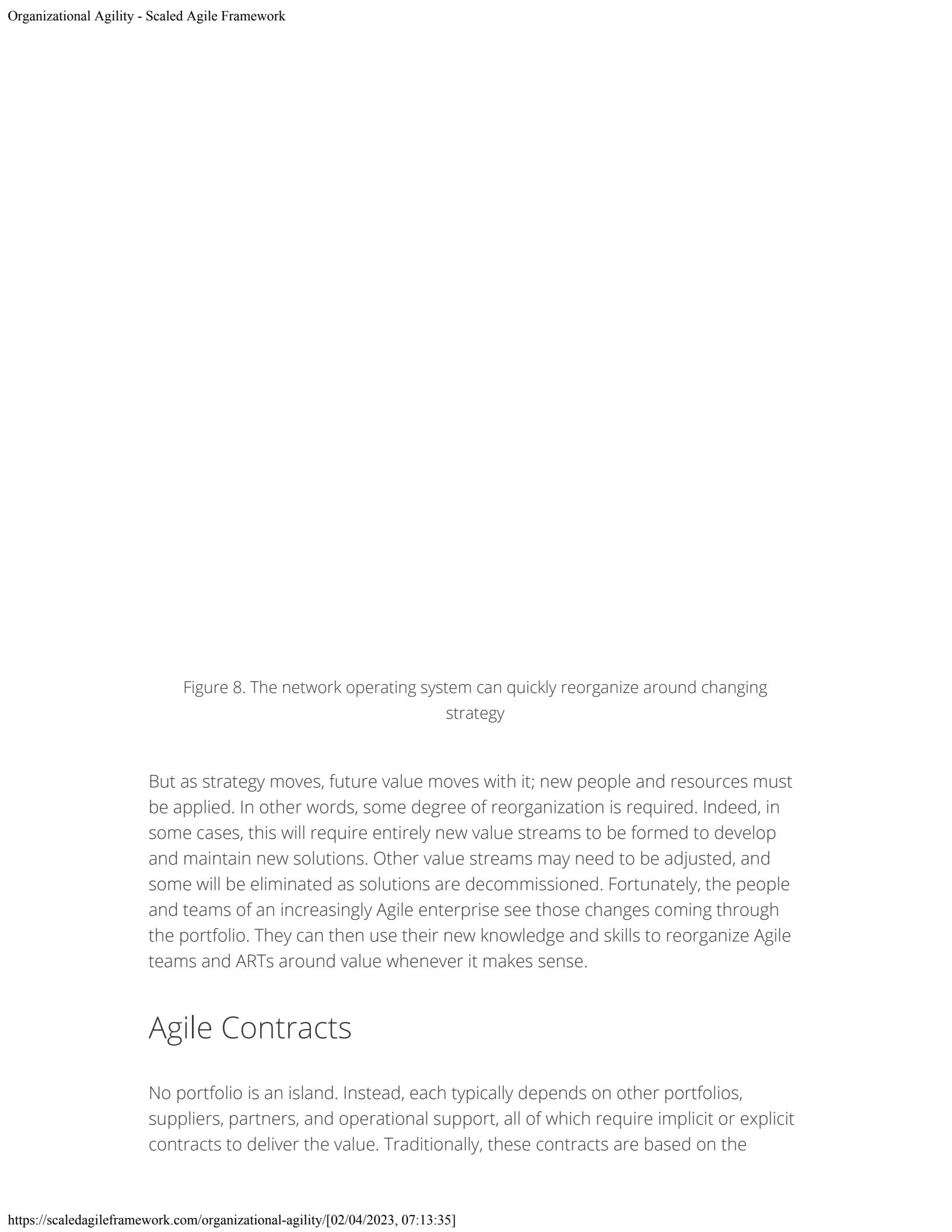


Figure 8. The network operating system can quickly reorganize around changing strategy

But as strategy moves, future value moves with it; new people and resources must be applied. In other words, some degree of reorganization is required. Indeed, in some cases, this will require entirely new value streams to be formed to develop and maintain new solutions. Other value streams may need to be adjusted, and some will be eliminated as solutions are decommissioned. Fortunately, the people and teams of an increasingly Agile enterprise see those changes coming through the portfolio. They can then use their new knowledge and skills to reorganize Agile teams and ARTs around value whenever it makes sense.

Agile Contracts

No portfolio is an island. Instead, each typically depends on other portfolios, suppliers, partners, and operational support, all of which require implicit or explicit contracts to deliver the value. Traditionally, these contracts are based on the

assumption that requirements, deliverables, and service levels are known upfront and will remain stable. We know from experience that it is just not true. As strategy changes, these traditional contracts can become enormous impediments that lock the business into assumptions of a former strategy. Although the business would like to change strategy, it is blocked by existing contracts.

Achieving business agility requires a more flexible approach to all types of contracts. How this is achieved depends on the nature of the contract, but each must be considered in terms of the adaptability required. Contracts for suppliers that provide components, subsystems, or services for enterprise solutions are particularly critical as they may tend to lock solution elements into requirements that were fixed long before. The [Agile Contracts](#) advanced topic article provides guidance on contracting strategies that can provide the needed flexibility,

Summary

Without organizational agility, enterprises cannot react fast when things happen. To be fully responsive to threats and opportunities requires Lean and Agile ways of working to spread throughout the entire organization. This change demands a workforce trained in Lean-Agile practices and understands and embodies the culture, values, and principles.

Lean business operations recognize that delighting customers goes further than purely solution development. The entire customer journey, which includes delivering, operating, and supporting business solutions, must be continually optimized to reduce time to market and increase customer satisfaction. Strategy agility provides the ability to sense and respond to market changes, evolve and implement new strategies quickly, and reorganize when necessary to address emerging opportunities. As a result, change becomes an opportunity, not a threat.

Learn More

[1] [SAFe for Marketing](#) whitepaper

[2] Hesselberg, Jorgen. *Unlocking Agility: An Insider's Guide to Agile Enterprise Transformation*. Addison-Wesley Signature Series (Cohn). Pearson Education, 2018.

[3] Kersten, Mik. *Project to Product*. IT Revolution Press, 2018.

[4] Martin, Karen. *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation*. McGraw-Hill Education, 2013.

[5] Chen, James. *Sunk Cost Dilemma*. Investopedia, 2022.
<https://www.investopedia.com/terms/s/sunk-cost-dilemma.asp>.

Last updated: 18 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Context is the keyÑ from that comes the understanding of everything.

ÑKenneth Noland

Solution Context

Solution Context identifies the critical aspects of the environment in which a solution operates.

Adjust Cookie Settings

It provides an essential understanding of the solutionÕs requirements, usage,

installation, operation, and support. Solution context heavily influences opportunities and constraints for releasing on demand.

Understanding the [Solution](#) context is crucial to value delivery. It impacts development priorities, [Solution Intent](#), [Capabilities](#), [Features](#), and [Nonfunctional Requirements](#) (NFRs). It provides opportunities, limits, and constraints for the [Continuous Delivery Pipeline](#) and other solution-level [Release on Demand](#) activities.

The solution context is often driven by factors outside the organization's control that develop the solution. The level of coupling between the solution and its context generally represents an architectural and business challenge: finding the right balance between flexibility and tightly coupled interaction with the environment—interactions that often cross internal, [Supplier](#), and [Customer](#) organizational boundaries. [Product](#) and [Solution Management](#) seek to influence the solution context in ways that benefit both the customer and the enterprise.

Details

Rarely are systems built for one's use; instead, they are made for others, whether internal operational [Value Streams](#) or external customers. This means no single person typically controls or fully understands the full context of the system's deployment and use. Instead, a method is shipped, deployed, installed, and maintained in an environment unlike that in which it was developed. Even in the case of internal IT systems, newly developed systems are typically hosted by the IT maintenance and operations teams. In this case, the production environment may differ from the development environment (see the [DevOps](#) article). Therefore, understanding the solution context is critical to reducing risk and achieving fitness for purpose.

Understanding and aligning the solution and solution intent with the solution context requires a customer-centric mindset. As Figure 1 illustrates, collaboration is needed. The level of cooperation depends heavily on the level of coupling between the solution and its environment.

Figure 1. Solution intent and context inform each other

Figure 1. Solution intent and context inform each other

To ensure this alignment, the customer (or a proxy) should participate in [PI Planning](#) ([Pre-Plan](#) and [Coordinate and Deliver](#) for Solution Trains) and [Solution Demos](#) as frequently as possible. As solutions increase in size and complexity, the customer should be increasingly involved in integrating the solution into their often unique and specialized context. Ideally, the customer is aligned with the release cadence of the ART, as cadence-based interaction and integration allow for building solution increments based on correct assumptions and provides validation of the result.

Solution Context Drives the Solution Intent

The customer's context drives requirements and puts constraints on design and implementation decisions. Many of these contextual requirements are non-negotiable (often caused by [Compliance](#) concerns) and may render the solution unusable if not included. These requirements fall under the fixed category of the solution intent. Many aspects of the solution context surface as [Nonfunctional Requirements](#) (NFRs) and need to be included as part of the definition of done for a solution increment.

The solution context may also require specific content that the solution intent must address. In a hierarchical system of systems, its intent may also be hierarchically dependent. The system context defines how the solution intent must be organized, packaged, and integrated for use by the customer to meet any compliance, certification, and other objectives.

Fixed versus Evolving Solution Context

Some solution contexts are established customer environments the solution must support, such as legacy systems containing critical customer data or older hardware models still in production. In that case, some or all of the previous and current solution context requirements are imposed on the solution via solution intent.

However, in many cases, new solutions may require the evolution of the customer's deployment environment. The changes must be actively tracked, as the system and deployment environment must evolve to a common state. In this latter case, fixed versus variable thinking and the preservation of options via multiple potentially viable solution contexts (see the "Moving from Variable to Fixed Solution Intent" section in [Solution Intent](#)) are tools to manage risk. Increasing variability and rapidly evolving solution contexts motivate more continuous collaboration.

Types of Solution Contexts

Understanding the customer's solution context helps determine how their system will be packaged and deployed in its ultimate operating environments. Examples of solution contexts might include environments such as:

- System of systems (ex., avionics system as part of the aircraft), product suite (ex., word processor as part of an office suite)
- IT deployment environments (ex., cloud environment where the solution is deployed)
- A single solution used in different usage models (ex., a single airliner that can fly both domestic and international routes economically)

Solution Context for a System of Systems

The solution supplier-to-customer relationship in significant system-of-systems contexts is typically unique and cascading, as Figure 2 shows.

Figure 2. Solution contexts wrap into a system of systems

Figure 2. Solution contexts wrap into a system of systems

Each organization in the supply chain delivers its solution to the customer's context, which specifies how the solution is packaged, deployed, and integrated. That customer, in turn, provides a solution in context to their customer, and so on. In Figure 2, for example, a vehicle navigation system supplier operates first in the infotainment supplier's context, then in the vehicle manufacturer's context, and finally in the consumer's context. All these contexts can impact the solution's viability, so one must know the entire end-to-end value chain.

Solution Context for IT Deployment Environments

The customer may also be internal when developing software solutions for internal use, but delivering solutions into the production environment still requires context. The deployment must consider specific interfaces, deployed operating systems, firewalls, APIs to other applications, hosted or cloud infrastructure, and so on, as Figure 3 shows.

Figure 3. Solution context for internal IT deployment

Figure 3. Solution context for internal IT deployment

Making deployment as routine as possible is the primary purpose of DevOps and the continuous delivery pipeline.

In this example, the new customer relationship management (CRM) system should reflect the required interfaces and how the application is packaged, released, hosted, and managed in the end environment.

Solution Context Includes Portfolio-Level Concerns

Generally, the products and services of an [Enterprise](#) must work together to accomplish the larger objectives of the solution. Therefore, most solutions are also a [Portfolio Level](#) concern. As such, emerging initiatives, typically in portfolio [Epics](#), drive solution intent and impact the solution's development and deployment.

For internally hosted systems, interoperability with other solutions is often required, further extending the solution context. For example, more significant operational value streams (see [Value Streams](#) article) often use solutions from multiple development value streams, as Figure 4 illustrates.

Figure 4. Solutions work together to support the full operational value stream

Figure 4. Solutions work together to support the full operational value stream

Each solution must collaborate and integrate to provide the operational value stream with a seamless, end-to-end solution.

Continuous Collaboration Ensures Deployability

Ensuring a solution will operate correctly in its context, and deployed environment requires continuous feedback (see the [Continuous Delivery Pipeline](#) article).

Cadence-based development frequently integrates the entire system-of-systems solution to demonstrate progress toward the top-level context's Milestone and release commitments. Continuous collaboration helps ensure that the solution can be deployed in the ultimate customer's context:

The customer raises and discusses context issues during PI planning and solution

demos

Solution Management and the customer continually ensure that the vision, solution intent, [Roadmap](#), and [Solution Train Backlog](#) align with the solution context

Issues discovered in the customer's context run through the Solution Train Kanban system for impact and resolution

Understanding and sharing relevant solution context knowledge, environment, and infrastructure, such as interface mock-ups, test and integration environments, and test and deployment scripts

The Solution Architect ensures technical alignment with the solution context—interfaces, constraints, etc.

Consequently, there are many collaboration points between the Agile teams and the various roles within the customer organization. Several SAFe functions carry that responsibility along with their customer counterparts, as shown in Figure 5.

Figure 5. Collaboration between SAFe and customer roles

Figure 5. Collaboration between SAFe and customer roles

Therefore, effective collaboration between customers and SAFe roles helps ensure

that the system meets the customers' needs in their context.

Last update: 7 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)
[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Nothing beats an
Agile Team.*

NSAFe mantra

Agile Teams

An Agile Team is a cross-functional group of typically ten or fewer individuals with all the skills necessary to define, build, test, and deliver value to their customer.

Agile Teams may be technical teams focused on building digitally-enabled solutions, business teams delivering business functions, or, increasingly, elements of both. By quickly delivering work in small increments, all Agile Teams strive for fast learning, gaining fast customer feedback, assessing the results, and adjusting

Adjust Cookies Settings

accordingly.

Details

Agile teams are self-organizing and self-managing and are accountable for delivering results that meet the needs and expectations of their customers and stakeholders.

Agile Teams power the [Agile Release Train](#) (ART) and thereby, the entire development portfolio. Agile teams collaborate with other teams to deliver ART solutions. They contribute to the [Vision](#) and [Roadmap](#), and participate in ART events. In addition, teams build the [Continuous Delivery Pipeline](#) (CDP) that accelerates the flow of value and supports the ability to [Release on Demand](#).

Agile teams are cross-functional, long-lived, and organized to deliver value as easily as possible. By building longer-lived teams and trains, enterprises can eliminate the start-stop-start ÔprojectÕ way of working (see [Lean Budgets](#)) and eliminate waste and delays in the process. Agile Teams [Lean-Agile Leaders](#) provide the vision, guidance, and autonomy necessary to foster and promote high-performing Agile teams. As a result, assigning work to individual team members is no longer required. Teams become self-directed and self-reliant, have more autonomy, further enabling [decentralized decision-making](#) all the way to the individual contributor. Agile teams are more productive than groups of similar individuals, are more engaged in their work, and have more fun on the job.

Characteristics of Agile Teams

All Agile teams share certain defining characteristics, as described in the following sections.

Teams Constitute the ART

Most Agile Teams are a part of an Agile Release Train and deliver value together with other teams that operate within the context of a common solution mission. They synchronize frequently with other teams, stakeholders, and their management. Some Agile teams—for example, business teams, enabler teams that support multiple ARTs, independent research teams, LACE teams, etc.—can deliver value independently of an ART, but they still benefit from their Agile method in establishing the flow of customer value.

This article describes common characteristics and responsibilities for all types of SAFe Agile teams.

Agile Teams are Cross-functional

Agile teams are composed of members dedicated full-time to their teams and contain all the functions they need to deliver value (Figure 1). This avoids individuals multiplexing across teams and eliminates the handoffs and delays that occur when pushing value across functional silos. Most generally, Agile Teams are capable, enabled, and able to:

Define Elaborate and design the features and stories needed to deliver customer value

Build ⚡ Contain all skills necessary to create the elements of the solution

Test ⚡ Ensure quality and performance of new the functionality

Deploy ⚡ Deploy increments of value to their customer

Figure 1. Agile teams are cross-functional

Figure 1. Agile teams are cross-functional

Agile Teams are Organized Around Value

SAFe Principle #10 [Organize around value](#), guides enterprises to organize people and teams around one goal: the continuous delivery of value to the customer. But to do so, they must consider how best to design their Agile Teams. As described in the book *Team Topologies* [1], SAFe recommends four primary ways to organize Agile Teams (Figure 2).

Stream-aligned teams are end customer-aligned and are capable of performing all the steps needed to build end-to-end customer value

Complicated subsystem teams are organized around critical solution subsystems. They focus on areas of high technical specialization, which limits the cognitive load on all the teams

Platform teams provide application services and APIs for stream-aligned teams to be able to leverage common platform services

Enabling teams provides tools, services, and short-term expertise to other teams

Figure 2. Applying team topologies to Agile teams on an ART

Figure 2. Applying team topologies to Agile teams on an ART

Further guidance on this important aspect of organizing Agile Teams can be found in the extended guidance article [Organizing Agile Teams and ARTs: Team Topologies at](#)

Scale.

High-Performing

Great teams require more than talented individuals. Team composition and dynamics play a significant role. In fact, *who* is on a team has less impact on performance than *how* the team works together. High-performing teams share many “teaming” characteristics:

- Alignment on a shared vision with clear goals and purpose
- A safe environment for taking risks without fear of embarrassment or criticism
- Diversity of knowledge and skills to independently make quick, effective decisions
- Mutual trust that allows for both healthy conflict and reliance on others
- Accountability to each other and the organization for reliably completing quality work
- Meeting commitments
- Understanding their work’s broader impact on the organization
- Having fun with their work and with each other

SAFe’s [Organizational Agility](#) competency provides more information on how Lean-thinking people and high-performing Agile Teams work to create better business outcomes.

Enabled by Critical Roles

Agile Teams are further enabled by two specialty roles (Figure 3).

1. The [Product Owner](#) contributes to the Vision and roadmap and works with the team to define Stories and prioritize the team’s work. By working with the customer and the teams, they define a backlog that addresses customer needs and also helps maintain the technical integrity of the product.
2. The [Scrum Master / Team Coach](#) (SM/TC) helps implement and maintain Agile practices, optimizes and improves team performance, partners with the [RTE](#) to guide improvements of the entire ART, and helps to optimize the flow of value.

Figure 3. Agile Teams include two special roles

Figure 3. Agile Teams include two special roles

When an Agile Team applies [SAFe Scrum](#), the SM/TC has specialty skills to facilitate effective implementation of SAFe Scrum. When a team applies [SAFe Team Kanban](#), the SM/TC has the specialty skills to facilitate an effective Kanban implementation.

Establishing Flow with Scrum and Kanban

Every Agile Team is responsible for establishing a fast, reliable flow of value to the customer. They achieve this by mastering two primary aspects (Figure 4):

1. A team operating model – SAFe Scrum or SAFe Team Kanban
2. The SAFe [Team Flow](#) accelerators that enhance the implementation of the model

Figure 4. Team flow is enabled by the team operating model and flow accelerators

SAFe Scrum and SAFe Team Kanban provide a set of practices that guide the team. This includes events, communication strategies, and specific rules that direct the progress of the work. But these methods work best with an underlying paradigm that helps the team maximize the flow of value to the customer. SAFe Flow Accelerators (SAFe Principle #6) provides this guiding paradigm. As part of this, teams:

- Work in small batches
- Keep work-in-process under control
- Address bottlenecks
- Periodically retrospect the product and the process

Most teams start their Agile journey by adopting SAFe Scrum. Practices like cadence-based planning, commitment to iteration goals, frequent retros, a daily sync, and adhering to a short iteration timebox are routine.

However, the work of some teams is better suited to respond to frequent and less plannable events. In this case, SAFe Team Kanban is often the preferred team operating model. SAFe Team Kanban is less dependent on iteration timeboxes, focusing more on a continuous flow of stories through the backlog to the customer.

Both methods are highly effective and are more alike than they are different. And in SAFe, both types of teams apply a Kanban system to manage their backlogs and work activities. In addition, many Agile teams build hybrid models to address their specific needs.

Responsibilities

The objective of every Agile Team is the same: to build the great products their customers need. They fulfill six primary areas of responsibility, as shown in Figure 5.

Figure 5. Areas of responsibility of an Agile Team

Figure 5. Areas of responsibility of an Agile Team

Each is described in the sections below.

Note: "Product" is an important choice of words here. Realistically, not every

team delivers a tangible, standalone product to an end-user customer, more typically, the full ART is required to do that. However, every team can and should recognize that whatever value they deliver—be it product, systems, subsystem, component, services, APIs, or other valuable assets—all benefit from treating their work as a product and knowing their customer, be they internal or external to the enterprise.

Connecting with the Customer

Agile Teams are responsible for understanding customer needs and defining the functionality needed to satisfy them. In order to develop a thorough understanding of the customer context, they apply [Customer Centricity](#). To understand the problem and design the right solution, they apply [Design Thinking](#). Doing so requires that all Agile teams:

Build empathy with the customer ☰ To build a great product for the customer, the team needs to think like the customer. However, often due to multiple degrees of separation from the customer, teams may struggle to understand actual customer needs and what represents value to them. Thus, increasing a team’s exposure to the customer context is essential. There are many ways to do that, including:
Leveraging the Product Owners’ skills, knowledge, and responsibilities
Establishing direct communication with the customer
Participating in solution support
Direct observation of the customer in action
Implementing solution telemetry to monitor usage

Additionally, effective Agile teams spend time developing and understanding their primary user personas—and their needs, struggles, and opportunities for improvement.

Participate in product definition ☰ Agile team members leverage their knowledge of customer personas to create user stories and acceptance criteria. While the solution vision and feature definition is led by [Product Management](#), it’s the teams that create the stories that fulfill that vision, as led by the Product Owner (PO).

Design and execute experiments ☰ As a part of customer and solution research, Agile Teams plan, execute, and review the results of various experiments. They implement research [spikes](#), low-fidelity models, and prototypes to gain fast feedback.

Planning the Work

Agile Teams plan their own work. Planning allows teams to stay aligned with the rest of the train and progressively refine work within a short timeframe. Planning involves all team members and relies on collaboration and transparency.

Effective planning facilitates alignment to a common goal while leveraging the flexibility and autonomy of each team member in achieving their objectives. Planning occurs at two levels:

ART Planning ☰ [PI Planning](#) is the event where each Agile Team gains alignment with the rest of the train and creates their backlog for the upcoming PI. PI Planning provides the larger, system view that is necessary to achieve a shared goal. As a result of PI Planning, the team creates a set of [PI Objectives](#) and a story-level outline of the planned progression of their work across iterations. This seeds the [Team Backlog](#) for the upcoming PI.

Team planning ☰ Once ART alignment has been established, teams perform shorter-term planning on a regular basis during the PI. The purpose of this planning is to leverage new learnings and plan the next short increment of value. The planning approach differs depending on whether a team applies SAFe Scrum or SAFe Team Kanban.

Refining the Team Backlog ☰ As knowledge emerges, teams continuously refresh and refine their backlog. The backlog is used to identify and prioritize the upcoming work they need to do to deliver their committed value.

Delivering Value

Value delivery is the primary task of an Agile Team. As a part of this effort, a team must be able to define, build, and test their stories. Many teams can also directly deploy new functionality into production or release them directly to the customer. This is the core process that takes place in the [Development Value Stream](#) to which the team contributes.

Frequently integrate and test ☰ A fast rhythm of development requires frequent integration and testing. This helps uncover technology and implementation problems early and gives the teams enough time to respond to the findings. The articles on [Build-in Quality](#) and [Team and Technical Agility](#) provide deeper guidance on these practices.

Regularly synchronize with the rest of the train ☰ While executing the PI, a team has multiple checkpoints with the rest of the train. This can take place in the form of an ART Sync that includes Coaches Sync and PO Sync. These events create visibility into the progress toward current PI objectives and help the ART make timely adjustments.

Build the continuous delivery pipeline ☰ An effective Agile development process also depends on a continuous delivery pipeline that has mechanisms for [Continuous Exploration](#), [Continuous Integration](#), and [Continuous Deployment](#). This typically requires value stream mapping to identify sources of delay and excessive variability.

Release frequently ☰ Some teams are able to release directly to the customer. These teams may typically in collaboration with some specialized teams or [Shared Services](#) establish their own release process. Decisions on when to release value are typically made at different levels: major releases may be decided upon during PI Planning; routine deployments are governed at the iteration level. Others can even be event-driven.

Getting Feedback

The speed of solution development depends directly on the speed and fidelity of the feedback the team can obtain. Without it, the team cannot adjust the course quickly. Errors start to accumulate, resulting in ineffective and delayed solutions. Both customer and technology feedback is needed to move forward effectively.

Find pathways to the customer ☰ In a large organization, the customer may be many degrees of separation away from the Agile team that creates value. The Product Owner serves as a local customer proxy and can be instrumental in helping the team establish the right connections to obtain direct customer feedback. [System Demos](#) are one productive venue for customer feedback. Teams should also seek feedback from ad hoc interactions with the customers who are using the solution in their working environment.

Frequently validate technical concerns ☰ A team must continually validate the assumptions behind the solution architecture and the implementation strategy. Technology feedback comes from frequent integration, testing, and deployment. Additionally, research spikes and prototypes help to cost-effectively explore technical strategies.

Improving relentlessly

Relentless improvement is a core value of SAFe. Agile teams constantly seek ways to improve their process and the outcomes they are responsible for.

SAFe provides a comprehensive approach for measuring

competency, flow, and outcomes, the three primary measures that predict business results (Figure 6).

Figure 6. The three areas of measurement

As a part of the improvement effort, the teams do the following:

Run routine improvement events ▷ Many teams use regular team-level retrospectives during iterations. Additionally, all ART teams participate in a joint **Inspect & Adapt** event with leaders whose help can be crucial in establishing and implementing necessary corrective action.

Improve some things immediately ▷ Some problems should be addressed as they occur, without waiting for the next improvement event. Addressing issues as they emerge is an integral part of a culture of continuous improvement.

Learn More

[1] Skelton, Matthew, and Manuel Pais. *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution Press, 2019.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Good design is actually a lot harder to notice than poor design, in part because good designs fit our needs so well that the design is invisible, serving us without drawing attention to itself. Bad design, on the other hand, screams out its inadequacies, making itself very

Adjust Cookie Settings

noticeable.

—Don Norman, *The Design of Everyday Things*

Design Thinking

Design Thinking is a customer-centric development process that creates desirable products that are profitable and sustainable over their lifecycle.

It goes beyond the traditional focus on the features and functions of a proposed product. Instead, it emphasizes understanding the problem to be solved, the context in which the solution will be used, and the evolution of that solution.

Note: This article focuses on the tools and practices associated with implementing design thinking. It should be read along with the [Customer Centricity](#) article, which focuses on the mindset and impact of customer centricity.

Details

Traditional waterfall approaches to product development are sequential: requirements are defined, and solutions are designed, built, and delivered to the market. The focus tends to be on the most apparent problems. Often, success is determined by implementing a solution that meets the *requirements* instead of the *user's needs*. *This results in products and services with unusable or ignored features that frustrate users and fail to meet the enterprise's business goals.*

Design thinking (Figure 1) represents a profoundly different approach to product and [Solution](#) development, in which divergent and convergent techniques are applied to understand a problem, design a solution, and deliver that solution to the market.

Figure 1. Design thinking activities

Figure 1. Design thinking activities

Design thinking also inspires new ways to measure the success of our efforts:

Desirable D Do customers and end-users want the solution?

Feasible D Can we deliver the right solution through a combination of build, buy, partner, or acquire activities?

Viable D Is the way we build and offer the solution creating more value than cost? For example, in a for-profit enterprise, are we profitable?

Sustainable D Are we proactively managing our solution to account for its expected product-market lifecycle?

Successive applications of design thinking advance the solution over its natural market lifecycle, as shown in Figure 2.

Figure 2. Advancing the solution over the lifecycle through design thinking

Figure 2. Advancing the solution over the lifecycle through design thinking

Understanding the Problem and Solution Space

In Figure 1, the core design thinking processes appear as a Ôdouble diamond.Õ This represents the focus on thoroughly exploring the problem space before creating solutions. Each diamond focuses on divergent thinking (understanding and exploring options) followed by convergent thinking (evaluating options and making choices).

The activities associated with exploring the problem are elaborated as follows:

Discover □ The discover phase seeks to *understand the problem* by engaging in market and user research to identify unmet needs. This research creates fresh perspectives that drive innovation. Unlike research that confirms or refutes a hypothesis, the inquiries associated with the discovery phase occur without preconceived notions about how users *should* work. Instead, it focuses on how users *work*. An essential research technique is Gemba, also known as Ôgoing to the place

where the work is done.◊

Define ◊ The define phase focuses on the information gathered during the discover phase to generate insights into specific problems and unmet needs. These create opportunities for the business and new product development. Results of this phase typically include personas and empathy maps (described below) that focus the product team on the solutions the *Customer* would view as desirable. *Epics* and *Features* capture the perceived changes needed for existing products and solutions.

With a clear understanding of the target market and its problems, the focus can move toward designing a solution, the second diamond of design thinking. These are:

Develop ◊ The develop phase uses journey mapping, story mapping, and prototyping to design potential solutions to problems quickly and cost-effectively. Each of these techniques is discussed more thoroughly later in this article. The develop phase also embraces *SAFe Principle #3* ◊ Assume variability; preserve options. Design thinking techniques preserve options responsibly.

Deliver ◊ The deliver phase produces artifacts suitable for creating the solution and varies based on context. These artifacts often start as prototypes expressed as validated features in the ART Backlog for continuous delivery.

Using Personas to Focus Design

Creating solutions for a *direct* customer—bespoke solutions—offer designers the advantage of speaking directly and frequently with a few target users, permitting them to participate in the design, *PI Planning*, *System Demos*, and other SAFe events. In some organizations, Customers are considered part of the team, so creating a Persona to represent them isn't typically needed but may be helpful when the organization is highly distributed.

In contrast, in an *indirect* customer market, which is common in B2C solutions, product teams need a way to maintain a connection with their target customers. So, they develop “personas,” fictional consumers and users derived from user research. [2] They depict the people who might similarly use a product or solution, providing insights into how real users would engage with a solution. User personas also support market segmentation strategy by offering a concrete design tool to reinforce that products and solutions are created for people. Personas drive product development and several SAFe practices, as shown in Figure 3.

Figure 3. How Personas can drive key activities in SAFe

In addition to user personas, *buyer* personas extend design thinking to include the individuals and organizations that authorize purchasing decisions. They help ensure that the design encompasses the whole product purchase experience, including after-sales service, support, and operations.

Establishing Empathy to Foster Customer-Centric Design

Customer-centric enterprises use empathy throughout the design process. Empathetic design dismisses preconceived ideas and uses the Customer's perspective to inform solution development.

Empathy maps [1] are a design thinking tool that promotes customer identification by helping teams develop a deep, shared understanding of others (Figure 4). They enable teams to imagine what a specific persona is thinking, feeling, hearing, and seeing as they use the product. The greater the degree of empathy a team has for its customers, the more likely it will be able to design desirable solutions.

Figure 4. Empathy map

Figure 4. Empathy map

Designing the User Experience through Journey Maps

A *customer journey map* illustrates the user experience in an [Operational Value Stream](#) that provides products and services. Figure 5 shows how these journey maps are powerful design thinking tools. They allow teams to identify ways the specific deliverables for one or more [Development Value Streams](#) can be improved to create a better end-to-end user experience.

Figure 5. Operational value streams and customer journey maps

Figure 5. Operational value streams and customer journey maps

Delivering Benefits Through Features

While a journey map captures the high-level experience of the Customer in the operational value stream, product features manage the specific deliverables that fulfill a stakeholder's need. Features are commonly described through a ***features and benefits (FAB) matrix***, using short phrases that provide context and a benefit hypothesis. Design thinking, however, promotes switching the order of the FAB to a ***benefit-feature matrix***. In this case, the intended customer benefits are identified first, and then the teams determine what features might satisfy their needs. This approach helps **Agile Teams** explore better and faster ways to deliver the desired benefits (Figure 6).

Figure 6. Design thinking promotes switching the order of the FAB to a *benefit-feature matrix*

Designing User Workflows or Journeys through Story Maps

Features that capture a workflow or user journey present a unique challenge to Agile teams. Because the backlog is a flat, one-dimensional list, it does not show the relationship between the user's goals, workflow activities, and the stories in the backlog. Story mapping is a brainstorming technique that can enable teams to design a solution focused on the Customer. Not all features will require story mapping. However, they are particularly useful for developing new end-user functionality for a workflow or customer journey.

Why Story Maps?

Story maps help teams ideate, plan, and group activities in a *workflow or user journey*. They allow teams to address the most critical steps before improving existing steps or adding new functionality. Story maps are an important design thinking tool that enables *Customer Centricity* because they focus on delighting a user instead of merely implementing stories ordered by their value. Another benefit is avoiding releasing a feature (or solution) that is not usable because its functionality depends on stories that are lower in priority and further down the backlog.

Figure 7 illustrates how a feature with a workflow is captured in a story map [3], organizing the sequence of stories according to the activities (or steps) a user needs to accomplish their goal. The first set of stories is essential for the initial release, while the next set represents improvements for future releases.

How to Create a Story Map

The following steps describe the process of creating a story map (Figure 7) for a new potential Feature that requires a workflow.

1. **Frame the purpose:** Identify the goal or customer problems the solution will solve and the intended users of the solution.
2. **Map the whole story:** Define the starting conditions for the user to accomplish their goals. Focus on describing the whole story and user activities and tasks, creating the backbone of the story map.
3. **Brainstorm:** Fill in the body of the story map by breaking down the larger user tasks into smaller subtasks and user interface details. Consider many possibilities without concern if the stories are in or out of scope. Affinity group the stories needed to complete the task under each activity.
4. **Identify the stories essential for the initial release:** The team identifies which stories can be released (in the next iteration or two) that will achieve a meaningful user outcome.
5. **Identify stories considered as improvements in future releases:** Stories that are not selected for the initial release will be added to the backlog as potential candidates for future releases.

Figure 7. Story map

Figure 7. Story map

Increasing Design Feedback Through Prototypes

A prototype is a basic functional model of a feature or product, usually built for demonstration purposes or as part of the development proc.s. It helps the team clarify their understanding of the problem and reduces risk in designing and developing the solution before making further investments. Prototypes provide many benefits:

Fast feedback. By definition, a prototype is cheaper and faster to produce than a complete solution. This enables faster feedback from users and customers, increased understanding of solution requirements, and greater confidence in the final designs.

Risk reduction. Prototypes can reduce technical risk by enabling Agile teams to

focus initial efforts on the aspects of the solution associated with the highest risk.

Intellectual property/patent filing. Prototypes can be used to satisfy strategic requirements for managing intellectual property as early as possible in the development process.

Models for requirements. Prototypes can provide more clarity in the requirements of the desired feature or solution than pages of documentation.

There are many kinds of prototypes, each optimized to provide different types of insights:

Paper prototypes are typically hand-drawn sketches of the intended solution. They can be automated to illustrate workflows or validate user story maps.

Mid-Fi prototypes are visually-complete representations of software-centric solutions but are not typically functionally integrated.

Hi-Fi prototypes are visually-complete and interactive models which users and customers can directly explore.

Hardware prototypes provide critical feedback on form factors, sizes, and operational requirements. For example, when exploring form factors to see how a new tablet might fit into existing backpacks, briefcases, and cars, one Silicon Valley company cut many plastic models from a single sheet of plastic. Later in this design process, this same team found they needed to redesign the power supply so that it would not unduly interfere with the WiFi signal.

Product teams should strive to leverage the lowest-cost, fastest form of prototyping to gain actionable feedback. Often, paper prototyping is the best choice. [4] [5]

Learn More

[1] Empathy Map Canvas. <https://medium.com/the-xplane-collection/updated-empathy-map-canvas-46df22df3c8a>

[2] Cooper, Alan, Robert Reimann, David Cronin, and Christopher Noessel. *About Face: The Essentials of Interaction Design 4th Edition*. Wiley, 2014.

[3] Patton, Jeff, and Peter Economy. *User Story Mapping: Discover the Whole Story, Build the Right Product* 1st Edition. O'Reilly Media, 2014.

[4] Snyder, Carolyn. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann, 2003.

[5] Gothelf, Jeff, and Josh Seiden. *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media, 2016.

Last updated: 13 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Continuous
attention to
technical excellence
and good design
enhances agility.*

NAgile Manifesto

Team and Technical Agility

Team and Technical Agility (TTA) is one of the seven core competencies of Business Agility. Each core competency is supported by a specific assessment, which enables the enterprise to assess its proficiency. The core competency assessments, along with the corresponding measures, are:

Measure

with recommended improvement opportunities, are available from the [and Grow](#) article.

Why Team and Technical Agility?

[Agile Teams](#) create and support the business solutions that deliver value to the enterprise's customers. Consequently, an organization's ability to thrive in the digital age is entirely dependent on the ability of its teams to deliver solutions that reliably meet customers' needs. Team and technical agility is the real cornerstone of Business Agility. It consists of three dimensions, as illustrated in Figure 1.

Figure 1. The three dimensions of team and technical agility

Figure 1. The three dimensions of team and technical agility

Agile Teams ▷ High-performing, cross-functional teams anchor the competency by applying effective Agile principles and practices.

Teams of Agile Teams ▷ Agile teams operate within the context of an [Agile Release Train](#) (ART), a long-lived team of Agile teams that provides a shared vision and direction and is ultimately responsible for delivering solution outcomes.

Built-In Quality ☰ All Agile teams apply defined Agile practices to create high-quality, well-designed solutions that support current and future business needs.

These three dimensions are complementary and dependent forces that organize and guide the people who power the value stream.

Agile Teams

The Agile team is the basic building block of Agile development and is the first dimension of the TTA competency. An Agile team in SAFe is a cross-functional group of 10 or fewer individuals who iteratively define, build, test, and deliver value to their customers. These teams have the authority and accountability to manage their work, which increases productivity and accelerates time-to-market.

Each Agile team adopts [SAFe Scrum](#), [SAFe Team Kanban](#), or a hybrid method to regulate their synchronization and delivery cadence. They apply the chosen method to manage a shared backlog, deliver incrementally, build necessary [Architectural Runway](#), and obtain frequent customer and stakeholder feedback.

An Agile team's primary objective is to deliver great products. As shown in Figure 2, they accomplish this by focusing on five key areas of responsibility.

Figure 2 - The Agile Team's five areas of responsibility

Figure 2. The Agile team's five areas of responsibility

Each of these areas of responsibility is summarized below.

Connecting with the customer ☰ Agile teams understand their customers' specific needs and use that knowledge to guide product development. They build empathy with customers and use that knowledge to collaborate with [Product Management](#) on product design and solution experiments.

Planning the work ☰ Agile teams plan their work and contribute directly to [PI Planning](#). They maintain their [Team Backlog](#), calibrate their priorities during Iteration Planning and Backlog Refinement, play a central role in PI Planning, and link their work to items in the [ART Backlog](#).

Delivering value ☰ Agile teams possess all the skills and resources required to deliver great products that delight their customers. They maintain alignment with the ART through various synchronization events and demos and establish [Continuous Delivery Pipelines](#) that enable them to frequently integrate, test, deploy, and release changes.

Getting feedback ☰ Agile teams seek feedback from customers regularly to understand the value of the products and technology delivered. They find pathways

to feedback that shorten the distance between them and their customers and continuously validate implementation decisions with other teams and architects.

Improving relentlessly ☰ Agile teams track the flow, competency, and outcome [metrics](#) that gauge their performance and contribute to business agility. They use these measurements to continually fuel retrospectives, define new performance targets, and implement improvements.

The Agile Manifesto guides Agile teams as they carry out their responsibilities (Figure 3). Although originally envisioned for software teams, these values apply to all types of Agile teams throughout technology and business.

Figure 3. The Agile Manifesto

Figure 3. The Agile Manifesto

Value delivery often spans traditional organizational boundaries. Therefore, Agile teams are multidisciplinary, containing all the people and skills needed to deliver value across functional domains, as depicted in Figure 4. Team members are dedicated full-time to the team, which creates a shared purpose and enhances flow.

Figure 4. Agile teams are cross-functional

Figure 4. Agile teams are cross-functional

Agile teams have two specialty roles (Figure 5). The **Product Owner** ensures that the Team Backlog is aligned with customer needs and guides the team toward delivering maximum business value. The **Scrum Master/Team Coach** is a servant leader and coach for the team, facilitating the agreed-to Agile process and fostering an environment that enables fast flow.

Figure 5. Agile teams include two specialty roles

Each Agile team maps to one of the following four topologies, specifying the team's role in the value stream.

1. **Stream-aligned team** ▷ An Agile team organized around the flow of work that can deliver value directly to the customer or end user.
2. **Complicated subsystem team** ▷ An Agile team organized around specific subsystems that require deeply specialized skills and expertise.
3. **Platform team** ▷ An Agile team organized around developing and supporting platforms that provide services to other teams.
4. **Enabling team** ▷ An Agile team organized to assist other teams with developing proficiency with new skills or technologies.

Teams of Agile Teams

Building enterprise-class solutions require more scope and breadth of skills than a single Agile team can provide. No one team can build and deliver large systems within a reasonable timeframe. And complex systems require a broad range of specialized skills that cannot be contained within a single team. Therefore, multiple Agile teams collaborate as members of an Agile Release Train, which operates across business functions to deliver one or more solutions (Figure 6).

Figure 7. Agile Release Trains build, deliver, and support valuable solutions

Figure 6. Agile Release Trains build, deliver, and support valuable solutions

Organized around one or more value streams, ARTs align teams to a common business and technology mission. They exist to deliver valuable solutions to their customers, which requires a broad spectrum of skills applied to five primary areas of responsibility (Figure 7).

Figure 8. Agile Release Train areas of responsibility

Figure 7. Agile Release Train areas of responsibility

The ART's areas of responsibility are identical to those of the Agile team. However, the ART applies them at scale, throughout the value stream, to deliver integrated solutions with compelling business value. And, like Agile teams, ARTs plan together, commit together, execute together, and improve together.

Built-In Quality

[Built-In Quality](#) is a set of practices that ensures that the outputs of Agile teams and ARTs meet appropriate quality standards. These practices favor building quality into products and services during development rather than simply inspecting for quality before release. Building quality in reduces the effort and cost associated with rework in the value stream, significantly accelerating value delivery.

As illustrated in Figure 8, built-in quality consists of several specialized practices organized into five primary domains. The remainder of this section summarizes

these elements.

Figure 9. Key domains and practices of Built-in Quality in SAFe

Figure 8. Key domains and practices of built-in quality in SAFe

Quality is evaluated and applied differently in different contexts. The same engineering standards would not govern a multi-player online game as a dialysis machine. Similarly, the development of a merger or acquisition agreement would be constrained by different quality requirements than an autonomous vehicle navigation system. SAFe organizes built-in quality practices into five domains to account for these major differences in context:

Business functions ▷ HR, marketing, finance, sales, and other disciplines that do not engineer technology-enabled solutions as a primary function are governed by quality standards that are unique to their fields. Examples could include accounting standards, marketing style guides, and employment laws.

Software applications ▷ Developing software-based solutions with built-in quality involves careful attention to functional, non-functional, and compliance requirements.

IT systems ▷ The infrastructure upon which software-based solutions operate must

be designed with the scalability, reliability, and security needed to provide sustained business value.

Hardware ☐ Unlike purely digital products, hardware-based solutions have physical features that must conform to specific weight, tension, heat, thrust, electrical, or other engineering specifications.

Cyber-physical systems ☐ Systems comprising a multitude of hardware components controlled by complex software routines—such as automobiles and aircraft—involve a complex array of quality standards that large teams of teams must address.

Working within these domains, Agile teams and ARTs apply specific practices early in the development cycle to ensure that quality is built into every product increment. Some practices apply across domains, while others are optimized for a single domain, as explained below.

Basic Agile quality practices ☐ These practices apply to all domains. They are the foundational building blocks of built-in quality and include techniques such as shifting learning left, pairing, peer review, collective ownership, and workflow automation.

Business quality standards ☐ These practices are unique to the *business functions* domain and typically address industry-specific standards and regulations outside IT. For example, legal teams may be bound by specific local and regional laws. HR teams may need to conduct business under strict privacy regulations.

Agile software development quality practices ☐ These practices apply to the development of any software-enabled solution. Continuous integration, test-first development, refactoring, and Agile architecture are some of the techniques in this category that enable built-in quality and streamline the delivery of digitally enabled solutions.

IT systems quality practices ☐ Implementing infrastructure-as-code, telemetry, observability, and automated governance, for example, helps build quality into the infrastructure that hosts the business’s applications.

Agile hardware engineering quality practices ☐ These practices apply to developing hardware-based solutions. The cost of change can be extremely high with these solutions, but modeling, simulation, and rapid prototyping help to assure quality and avoid expensive rework.

Cyber-physical systems quality practices ☐ Building quality into many of the world’s most complex systems is not trivial but is made easier with model-based systems engineering (MBSE) and frequent, end-to-end system integration.

These practices are mixed and matched appropriately to ensure all elements of a solution are developed with high quality and can be delivered in the shortest sustainable lead time. More detail can be found in the [Built-In Quality](#) article.

Accelerating Flow

As explained by [Principle #6 D Make value flow without interruptions](#), SAFe is a flow-based system that promotes fast, efficient value delivery. Flow is present when Agile teams, ARTs, and the portfolio can deliver high-quality products and services with minimal delay.

To accelerate flow throughout the value stream, wasteful activities are continuously detected and remediated at all levels of SAFe. In particular, the TTA competency enables Agile teams to quickly identify inefficiencies in their part of the value stream that can then be corrected using the flow accelerators described in the [Team Flow](#) article. The more proficient teams become with TTA, the faster they detect and correct flow issues.

TTA also contributes to an ART's ability to accelerate flow; however, the [Agile Product Delivery](#) competency enables [ART Flow](#) more directly.

Summary

TTA enables cross-functional Agile teams to accelerate value delivery without being bound to any specific way of working. This promotes long-lived teams that learn together, grow together, and deliver great products. ARTs comprise several Agile teams, are too cross-functional and can cover the complete solution delivery life cycle. Agile teams and ARTs are mutually responsible for building quality into everything they deliver. The result is an organization with the team and technical agility required to deliver continuous value at an enterprise scale.

Learn More

[1] Manifesto for Agile Software Development. [AgileManifesto.org](https://agilemanifesto.org)

Last update: 14 March 2023

images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Specifically, you can take the time to develop and bring to the table an outside-in, market-centric perspective that is so compelling and so well informed that it can counterbalance the inside-out company-centric orientation of last year's operating plan.

Adjust Cookie Settings

Agile Product Delivery

Agile Product Delivery is one of the seven core competencies of SAFe, which is essential to achieving [Business Agility](#). The [Measure and Grow](#) article provides a self-assessment for each competency, including APD, to evaluate a team's proficiency and identify improvement opportunities.

Why Agile Product Delivery?

Achieving business agility requires [Agile Teams](#) and [Agile Release Trains \(ARTs\)](#) to increase their ability to deliver innovative products and services rapidly. This capability requires balancing the focus on execution and customers, ensuring the creation of the *right solutions*, for the *right customers*, at the *right time*.

Figure 1 illustrates the three dimensions of APD:

1. Customer Centricity and Design Thinking
2. Develop on Cadence, Release on Demand
3. DevOps and the Continuous Delivery Pipeline

The mutually supportive capabilities of the APD competency create opportunities for sustained market and service leadership.

Figure 1. Three Dimensions of APD

Figure 1. Three Dimensions of APD

The following sections describe each dimension of APD.

Customer Centricity and Design Thinking

Customer centricity and design thinking comprise the first dimension of APD. This mindset and way of doing business put the customer first, at the enterprise's core, to provide positive customer experiences and build long-term relationships.

Customer Centricity

Customer Centricity is a mindset and way of doing business that focuses on creating positive user experiences and customer engagement with the organization's products and services. It puts the customer at the center of every decision, deeply considering the effect on its end users [1]. This mindset motivates long-term customer relationships, enabling more customer value, often in unexpected ways

[2]. This dimension of APD encourages Agile Teams to:

Focus on the customer **D** Employ user and market research, including developing personas, to align and focus the organization on specific, targeted user segments.

Understand the customer's needs **D** Invest time in identifying and building solutions that address these needs.

Think and feel like the customer **D** Apply empathy and strive to see the world from their customers' point of view.

Build whole product solutions **D** Design whole product solutions for the user's needs, ensuring that the initial and long-term user experiences are optimal and evolve as needed.

Create customer lifetime value **D** Move beyond a transactional mentality and focus on the overall customer relationship over the solution's life. [2].

Customer-centric businesses create greater profits, employee engagement, and customer satisfaction. Customer-centric governments and nonprofits create resilience, sustainability, and the alignment needed to fulfill their mission.

[Product Management](#) is responsible for coordinating and bringing new solutions to market while ensuring the ongoing success of existing products.

Design Thinking

[Design Thinking](#) is integral to *customer centricity*. It's an iterative development process that ensures solutions are desired by customers and users while also ensuring the solution is feasible, economically viable, and sustainable throughout its lifecycle.

It has two main activities that result in a sustainable solution:

Understanding the problem **D** The problem space is where designers explore the problem, including its complex nature, and get a clear definition of the problem, gaining insight into the requirements and benefits of a desirable solution

Designing the right solution **D** The solution space is where ideas are generated, visualized, and prototypes are developed and tested.

Figure 2 shows the core processes of design thinking, illustrated as a *double diamond*. This process focuses on thoroughly exploring the problem space before creating solutions.

Applying design thinking during development ensures the solution is *desirable*, *viable*, and *feasible*. At the same time, understanding and managing solution economics results in a *sustainable* product or service.

Figure 2. Design thinking process and activities

Figure 2. Design thinking process and activities

Understanding the problem typically includes the following two activities:

Discover D Seeks to understand the problem by engaging with users and market research to identify unmet needs.

Define D Analyzes the discover phase data using convergent techniques to generate insights into the specific problems and unmet needs.

After exploration, the organization has the inputs to begin designing a solution, which often involves the following activities:

Develop D Applies customer journey maps and story mapping to quickly design potential, cost-effective solutions.

Deliver D Produces various artifacts that are suitable for creating the solution. These solutions often start as prototypes with continuous delivery from the ARTs.

Figure 2 also illustrates how divergent and convergent thinking are applied to exploring ideas, working towards goals, and addressing challenges. Both are

necessary, and together they lead to unique solutions for challenges that require exploration and creativity.

Lean UX

In SAFe, [Lean UX](#) extends the traditional user experience design process beyond merely executing design elements and anticipating how users might interact with a system. Instead, it encourages a far more comprehensive view of why a [Feature](#) exists, the functionality required to implement it, and a *hypothesis* for its intended benefits. Leading indicators and getting immediate feedback from customers and end-users help determine if the system meets customer needs and business objectives. Lean UX provides a closed-loop method for defining, hypothesizing, building, measuring value, and learning.

In Lean UX, the designer's role evolves more toward design facilitation and taking on a new set of responsibilities. Besides Lean Startup, Lean UX has two other foundations: Design thinking and Agile development. Design thinking helps widen the scope of user experience work beyond mere interfaces and artifacts. It looks at the whole system and applies design tools to broader customer problems, relying heavily on collaboration, iterative approaches, and empathy as its core to problem-solving. [3]

Develop on Cadence, Release on Demand

Figure 3 illustrates the concept of developing on cadence and releasing on demand. It separates the concerns of *developing solutions* and *releasing* value, ensuring customers can get what they need when desired, which improves business agility.

Figure 3. Developing on cadence enables value to be released on demand

Figure 3. Developing on cadence enables value to be released on demand

Why Develop on Cadence?

In a flow-based system, establishing routine development activities on a fast, synchronized PI cadence—[a regular predictive rhythm of team and ART events](#)—is a proven strategy to manage the *inherent variability* in product development. The following activities support this cadence:

ART events □ The ART has several important cadence-based events: [PI Planning](#), [System Demos](#), and [Inspect and Adapt](#). The PO and Coach Sync events are held throughout the PI to help eliminate impediments, remove bottlenecks, and communicate adjustments needed by the teams.

Agile Team events □ PIs are divided into iterations, which help align Agile Teams and enable faster response to change. Team cadence-based events further support teams: [Iteration Planning](#), Team Syncs (usually held daily), [Iteration Review](#), and [Iteration Retrospective](#).

Simply put, teams use a process optimized for highly variable knowledge work, providing a reliable series of events and activities on a regular, predictable schedule.

Why Release on Demand?

[Releasing on demand](#) provides a significant strategic advantage by making value available when customers, the market, and the business need it. In collaboration with stakeholders, Product Management determines when a release should happen, what elements should be released, and who should receive it.

Some products serve market segments that offer new functionality as soon as it's available. While others may have distinct market rhythms that govern optimal release windows, as described in the [Roadmap](#) article.

Figure 4 illustrates the RoD processes by which new functionality is deployed into production and *released* incrementally or immediately to customers based on user or market demand.

Figure 4. The four activities of Release on Demand

Figure 4. The four activities of Release on Demand

Release describes the practices necessary to deliver the solution to end users, all at once or incrementally

Stabilize and Operate describes the practices needed to make sure the solution is

working well from a functional and non-functional perspective

Measure describes the practices to quantify if the newly-released functionality provides the intended value

Learn describes the practices needed to decide what should be done with the information gathered and prepare for the next learning loop through the CDP

Building and maintaining a [Continuous Delivery Pipeline](#) (CDP) allows each ART to define, build, validate, and release new functionality to meet their PI objectives.

DevOps and the Continuous Delivery Pipeline

[DevOps](#) and the Continuous Delivery Pipeline lay the foundation that enables releasing value, in whole or part, at any time to meet demand.

While releasing on demand is a goal of the CDP, gaining the competency to reliably and skillfully release value whenever desired is hard work. It involves embracing the DevOps mindset and culture and creating an increasingly automated pipeline.

Each ART builds and maintains (or shares) a CDP with the assets and technologies needed to deliver solutions as independently as possible. The first three aspects of the pipeline, [Continuous Exploration](#), [Continuous Integration](#), and [Continuous Deployment](#), support the delivery of new functionality, as illustrated in Figure 5.

Figure 5. The Continuous Delivery Pipeline

Figure 5. The Continuous Delivery Pipeline

Continuous Exploration promotes innovation and aligns with what should be built.

Design Thinking continually explores customer and market needs, defining a [Vision](#) and Roadmap.

Continuous Integration builds quality into the development process by continuously integrating many Agile Teams' work.

Continuous Deployment represents the processes associated with migrating solutions from staging to production.

As noted earlier, Release on Demand (Figure 4) is the ability to make value available to customers all at once or in an ad hoc fashion based on market and business needs.

Embracing DevOps Mindset, Culture, and Practices

High-performing organizations use DevOps to dramatically outperform competitors by delivering and supporting their products and services to respond to customer demands faster.

Figure 6 illustrates that Dev is frequently in a *fast-forward* mode, trying to keep

pace with the constant demand for changes and innovation. At the same time, Ops often presses **pause** on changes because they are accountable for production stability and resiliency.

DevOps aligns efforts across development, operations, and other business functions to achieve an optimal balance of speed and stability.

Figure 6. DevOps fosters collaboration across all functions

Figure 6. DevOps fosters collaboration across all functions

Ultimately, DevOps is a mindset, a culture, and a set of technical practices that provides solution elements to the customer without handoffs or too much external production and operations support. As illustrated in Figure 7, SAFe's approach to DevOps is grounded in five concepts: Culture, Automation, Lean Flow, Measurement, and Recovery (**CALMR**), briefly described below.

Figure 7. SAFe's CALMR approach to DevOps

Figure 7. SAFe's CALMR approach to DevOps

Culture ☰ A culture of shared responsibility is needed for fast value delivery across the entire Value Stream. All relevant departments help create value, including development, testing, security, compliance, operations, architecture, and more.

Automation ☰ Automation is used to reduce or eliminate human intervention from the CDP to decrease errors and reduce the overall cycle time of the release process.

Lean flow ☰ Fosters limiting work in process (WIP), smaller batches, and reducing queue lengths. In other words, they [Make Value Flow without Interruptions \(Principle #6\)](#) and enable faster customer feedback.

Measurement ☰ Supports learning and continuous improvement by understanding and measuring the flow of value through the pipeline.

Recovery ☰ Builds systems that allow fast fixes of production issues such as automatic rollback and “fix forward” capabilities, immutable infrastructure, and more.

Cloud Computing is a Key Enabler of DevOps

The ever-expanding universe of cloud capabilities has fundamentally changed how digitally enabled solutions are built, deployed, and maintained. Cloud computing has been one of the most disruptive drivers for changing the delivery model for enterprise IT since its inception [1]. Not surprisingly, the primary reason to move to the cloud is to increase product development speed and agility [2].

The cloud is everywhere, and it fuels digital business and enables DevOps and a more efficient CDP. SAFe enterprises can harness the power and ubiquity of the

cloud to increase agility in all areas of the organization.

Team and ART Flow

Since SAFe is a flow-based system, any interruptions to flow must be resolved quickly to enable continuous value delivery. SAFe provides six articles to help fix impediments to flow: Principle #6 – Make value flow without interruptions, [Value Stream Management](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#). Each article defines a set of eight flow accelerators that help identify, fix, optimize, and debug issues to achieve a continuous value flow.

The ART and Team flow guidance apply directly to the APD competency:

ART Flow – This represents a state where an ART delivers a continuous flow of value to the customer. It describes how teams-of-agile teams (ARTs) work with their stakeholders to get closer to customers and build CDPs. The CDP accelerates the delivery of products and services.

Team Flow – This represents the state in which *Agile Teams* deliver a continuous flow of customer value. The SAFe [Team and Technical Agility](#) (TTA) competency offers practices for creating effective cross-functional Agile Teams and ARTs. It fosters applying built-in quality practices and collaborating with extended stakeholders to deliver solutions faster.

Summary

Businesses need to balance their execution focus with a customer focus to help ensure that they are creating the right solutions, for the right customers, at the right time. APD is grounded in customer-centricity, design thinking, and Lean UX putting the customer at the center of every decision. It applies design thinking to ensure the solution is desirable, feasible, viable, and sustainable.

Developing on cadence helps manage the variability inherent in product development. Release on demand separates the release and development cadences to ensure customers can get what they need when needed. DevOps and the CDP create the foundation that enables releasing value, in whole or part, at any time to meet customer and market demand. APD enhances business agility, offering superior outcomes for the enterprise and its customers.

Learn More

[1] Norman, Don. *The Design of Everyday Things*. Basic Books, 2013.

[2] Osterwalder, Alexander, Yves Pigneur, Gregory Bernarda, and Alan Smith. *Value Proposition Design: How to Create Products and Services Customers Want*. Wiley, 2014.

[3] Gothelf, Jeff, and Josh Seiden. *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media, 2016.

[4] Lean UX vs. Design Thinking. <https://uxplanet.org/lean-ux-versus-design-thinking-3f9ebb8aef59>

Last updated: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Most strategy dialogues end up with executives talking at cross-purposes because É nobody knows exactly what is meant by vision and strategy, and no two people ever quite agree on which topics belong where. That is why, when you ask members of an executive team to describe and explain the corporate

Adjust Cookie Settings

*strategy, you
frequently get wildly
different answers.
We just don't have
a good business
discipline for
converging on issues
this abstract.*

~Geoffrey Moore,
Escape Velocity

Lean Portfolio Management

The Lean Portfolio Management (LPM) competency aligns strategy and execution by applying Lean and systems thinking approaches to strategy and investment funding, Agile portfolio operations, and governance.

LPM provides an alignment and governance model for a specific SAFe Portfolio, which contains a set of [Development Value Streams \(DVS\)](#) for a business domain in an [Enterprise](#). Each DVS builds, supports, and maintains [Solutions](#) delivered to the Customer, whether internal or external to the enterprise. Examples include developing an e-commerce website, medical device, or satellite and developing and deploying a software application within an enterprise for internal customers.

LPM is one of the seven core competencies essential to achieving [Business Agility](#). Each core competency is supported by a self-assessment, enabling the enterprise to assess its proficiency. The [Measure and Grow](#) article provides a competency assessment and recommends improvement opportunities for implementing LPM.

Why Lean Portfolio Management?

Traditional approaches to portfolio management were not designed to compete in the age of software and digital. Enterprises face a higher degree of uncertainty and need to deliver innovative solutions faster. Many legacy portfolio practices remain despite massive market changes and how businesses operate in the digital era.

Modernizing portfolio management is critical to supporting SAFe's Lean-Agile way of working and competing in this new reality. Fortunately, many enterprises have

already traveled this path, and the change patterns are apparent, as shown in Figure 1.

Figure 1. Critical shifts in portfolio mindset and practice

Figure 1. Critical shifts in portfolio mindset and practice

LPM has the highest decision-making and financial accountability for the solutions and development value streams in a SAFe portfolio. The people who fulfill the LPM function have various roles and titles and often reside in different parts of the organization's hierarchy. Because LPM is vital to the enterprise, executives and business owners who understand the financial, technical, and business contexts hold strategy and investment funding responsibilities. They are accountable for the overall business outcomes and addressing the challenge of defining, communicating, and aligning strategy with execution.

Figure 2 illustrates the three dimensions of LPM, followed by a brief description and a set of roles needed for its responsibilities.

Figure 2. The three dimensions of LPM

Figure 2. The three dimensions of LPM

1. **Strategy & Investment Funding** ensures the entire portfolio is aligned and funded to create and maintain the solutions needed to meet business targets.
2. **Agile Portfolio Operations** coordinates and supports decentralized ART execution and fosters operational excellence.
3. **Lean Governance** supports oversight of spending, audit, compliance, expenditure, measurement, and reporting.

The following sections describe these dimensions.

Strategy and Investment Funding

Strategy and investment funding ensures that the entire portfolio is aligned and funded to create and maintain the solutions needed to meet business targets. An enterprise can only accomplish its ultimate business objectives by allocating the Ôright investmentsÕ to building the Ôright things.Õ

However, portfolio strategy is much more than prioritization and selecting the best investments. The portfolio needs to understand its role in achieving the enterprise strategy. Therefore, LPM should understand the portfolio's current state, develop a plan to evolve to a better, differentiated future state, and continuously adjust the vision and plan to address the changing business context.

Some changes will require large initiatives to achieve the future state. LPM and its stakeholders need to create business and enabler Epics that feed directly into the [Portfolio Backlog](#).

The strategy and investment funding responsibilities (Figure 3) require collaborations among enterprise executives, [Business Owners](#), and [Enterprise Architects](#), including portfolio stakeholders and other technologists. Each responsibility is described in the sections that follow.

Figure 3. The strategy and investment funding collaboration and responsibilities

Figure 3. The strategy and investment funding collaboration and responsibilities

Connect the Portfolio to the Enterprise Strategy

The portfolio strategy should support the enterprise's broader business objectives. That's why connecting the portfolio to the enterprise strategy is the primary responsibility of the strategy and investment funding collaboration.

Moreover, linking the portfolio to the organization's strategy is bi-directional. The portfolio connects to the enterprise business strategy through [Strategic Themes](#) and the portfolio budget. It provides feedback to the enterprise via the portfolio context (described in the [Enterprise](#) article).

Maintain a Portfolio Vision

Portfolio Vision

The [Portfolio Vision](#) describes the future state of its development value streams and solutions. The current and future state difference represents the gap that LPM translates into the vision.

Effective collaboration across the portfolio requires the continuous communication of the portfolio vision, goals, ideas, and expectations openly and transparently. Business Owners should frequently communicate the vision and the strategic themes, for example, during PI planning, all-hands company meetings, and throughout the PI.

Gaining portfolio alignment requires people to work toward a shared goal and purpose beyond any individual, team, ART, or value stream. This *One portfolio mindset* is critical to achieving success toward the strategic themes and having the agility to make mid-course corrections fluently. The quote below reminds us that the purpose of alignment is not about centralized control.

*The more alignment you have, the more autonomy you can grant*Ñ

Stephen Bungay, author and strategy consultant

Instead, alignment provides a powerful way to turn the portfolio vision into an executable strategy. Indeed, this goes well beyond the words in a slide deck. It unleashes all portfolio members' creative and productive energy toward achieving its goals and further unlocks people's intrinsic motivation. Developing the vision and strategy can be difficult and time-consuming. However, this work is not a once-and-done exercise. As new information is learned about the solution set, including customer feedback and [Key Performance Indicators \(KPIs\)](#), the LPM function periodically reviews the portfolio canvas (e.g., quarterly). They explore scenarios where the portfolio could evolve to a better differentiated future state aligned with the strategic themes.

Enterprise Architecture

The road to a better future state should be paved with architectural principles and practices that enable the ongoing evolution of the portfolio's solution set. This makes enterprise architecture a critical component of strategy and investment funding. [Enterprise Architects](#) help translate the business vision and strategy into effective technology plans. Enterprise Architects promote adaptive design and engineering practices to drive the portfolio's architectural initiatives. Enterprise Architects also facilitate the reuse of hardware and software components and proven design patterns across the portfolio to help value streams develop and enhance solutions faster and with higher quality.

Organizations must respond simultaneously to new business challenges and implement larger-scale architectural initiatives requiring intentionality and planning. Emergent design alone is not sufficient at scale. Enterprise Architects help improve results by offering architectural governance and fostering the right balance between intentional and emergent design. Achieving this balance is essential to maintaining a healthy [Architectural Runway](#) across the portfolio and developing large-scale systems effectively. Therefore, Enterprise Architects may recommend technology standards and stacks for value stream level interoperability, APIs, hosting, and design methods for developing and testing cyber-physical systems.

Portfolio Roadmap

The best way to predict the portfolio's future state is to create it through a purposeful and flexible portfolio roadmap (Figure 4). Because some portfolio initiatives may take years to develop and are safety critical (for example, aerospace, autonomous vehicles, and cyber-physical systems), a larger planning horizon beyond a few PIs may be required. The portfolio and solution roadmaps are bi-directional; each roadmap influences the other.

Figure 4. The portfolio roadmap communicates the longer-term picture

Figure 4. The portfolio roadmap communicates the longer-term picture

The portfolio roadmap integrates lower-level roadmaps into a more comprehensive view. The initiatives in the portfolio roadmap influence the direction and timing of the solution roadmaps, as Figure 4 illustrates.

Since the portfolio roadmap may span multiple years, estimating longer-term initiatives requires Agile methods. However, every enterprise should be cautious about such forecasts. While long-term predictability is a worthy goal, use ***flexible rolling-wave*** roadmaps to replace fixed plans. Lean-Agile Leaders should know that ***every long-term commitment decreases the organization's agility.***

Realize Portfolio Vision Through Epics

Many vision changes will require large initiatives (**Epics**) to achieve the future state. Business epics directly deliver business value, while Enabler epics advance the architectural runway to support upcoming business or technical needs. Since epics often have lots of uncertainty, it's good practice to use the SAFe Lean Startup

Cycle for their implementation.

Understanding the epics' forecasted costs and gaining a high-level view of when the potential new value can be delivered is essential for comparing investments (see the portfolio roadmap described earlier). This forecast includes the MVP, proving or disproving the epic's hypothesis and implementation if LPM makes a persevere decision. Sometimes overlooked, contractors and Suppliers should be part of the cost equation.

Establish Lean Budgets and Guardrails

[Lean Budgets](#) and [Guardrails](#) offer funding and governance practices that improve development throughput while maintaining financial and fitness-for-use governance. This new funding model allows the enterprise to eliminate or reduce the need for traditional project-based funding and cost accounting, reducing friction, delays, and overhead. Lean budgets provide funding for value streams aligned with the business strategy and current strategic themes. Guardrails support these budgets by providing governance and spending policies and practices.

Establish Portfolio Flow

Portfolio business and enabler epics, capture, analyze, and approve new business and technology initiatives. These initiatives typically require the collaboration of multiple value streams or the creation of new value streams or ARTs.

In the context of strategy and investment funding, portfolio flow describes the process of managing portfolio epics through their lifecycle, including limiting the number of significant and typically cross-cutting initiatives in progress to match the portfolio's capacity. LPM uses the portfolio Kanban system to visualize and restrict work-in-process (WIP), reduce batch sizes, and control the length of longer-term development queues. Successfully establishing portfolio flow requires knowing the total capacity for new development work versus ongoing maintenance and support activities. The enterprise can objectively evaluate and originate portfolio-level initiatives only when this balance is understood.

Another aspect of portfolio flow concerns operational excellence, which is the subject of the next section, Agile Portfolio Operations.

Agile Portfolio Operations

Agile portfolio operations coordinate and support decentralized ART execution and enable operational excellence. SAFe principles and the Lean-Agile mindset foster

the *decentralization* of strategy execution to empower [Agile Release Trains](#) (ARTs) and [Solution Trains](#).

The Agile portfolio operations collaboration and responsibilities (Figure 5) require the active engagement of the Value Management Office (VMO), Lean-Agile Center of Excellence (LACE), Release Train Engineer (RTE), and Scrum Master/Team Coach CoP. Figure 5. illustrates each of these responsibilities, followed by a description of each.

Figure 5. Agile portfolio operations collaboration and responsibilities

Figure 5. Agile portfolio operations collaboration and responsibilities

Coordinate Value Streams

Although many value streams operate independently, cooperation among solutions can provide unique, differentiating portfolio-level capabilities and benefits that competitors can't match. [Value Stream coordination](#) defines how to manage dependencies and exploit the opportunities that exist only in the interconnections between value streams. To this end, Lean-Agile leaders understand their value streams' challenges and opportunities. They make them as independent as

possible while simultaneously interconnecting and coordinating them with the enterprise's larger purpose.

Support ART Execution

The LPM function can help cultivate and apply successful *ART execution patterns* across the portfolio with the assistance of the [Lean-Agile Center of Excellence](#) (LACE). The LACE is often responsible for leading operational excellence with the help of an *RTE and Scrum Master/Team Coach* CoP. Together, they can optimize, address, and debug issues from Agile Teams, ARTs, and value streams. The LACE and CoPs provide a forum for sharing effective Agile ART execution, flow practices, and other knowledge. They become a continuous energy source to power the enterprise through the necessary organizational changes.

Foster Operational Excellence

Operational excellence focuses on continually improving efficiency, practices, and results to optimize business performance. LPM plays a leadership role in operational excellence, helping the organization achieve its business goals.

Value Stream Management

[Value Stream Management \(VSM\)](#) is a leadership and technical discipline that enables the maximum flow of business value through end-to-end solution delivery.

Lean thinking is the foundation of Value Stream Management. The lean principles provide a shared mindset for everyone involved in solution delivery to improve operational efficiency and eliminate delays. Although everyone in a SAFe portfolio plays a role in VSM, Lean Portfolio Management (LPM) is accountable for establishing the value streams and fostering operational excellence.

The Lean-Agile Center of Excellence (LACE)

Operating under the auspices of LPM, the LACE also plays a significant role in fostering operational excellence. This typically includes:

- Facilitating Value Stream identification workshops
- Communicating the business need for SAFe
- Integrating SAFe practices and fostering [Communities of Practice](#)
- Creating alignment around organizational changes
- Providing coaching and training to ART stakeholders, Solution Trains, and Agile Teams
- Establishing objective measures for progress, product, and process (see PI milestones in the [Roadmap](#) article)

From PMO to VMO

Many enterprises have discovered that centralized decision-making and traditional mindsets can undermine the move to Lean-Agile practices. As a result, some enterprises have abandoned the PMO approach, distributing all the responsibilities to ARTs and Solution Trains. Unfortunately, this choice can inhibit the adoption of successful execution patterns, standard measures, and reporting that can be developed and applied across the portfolio.

One option is redesigning the traditional PMO to become *a Value Management Office (VMO)*. Operating through LPM, the VMO leverages the specialized skills, knowledge, and relationships of the current PMO while transitioning themselves and the portfolio to a new Lean-Agile way of working. VMO activities often include the following:

- Facilitates the portfolio events
- Works with the LACE to develop, harvest, and apply successful ART execution patterns across the portfolio
- Facilitates Lean budgeting and coordinates portfolio governance
- Fosters decentralized PI Planning and operational excellence
- Establishes objective metrics and reports progress toward business agility
- Focuses the portfolio on measuring and improving value delivery
- Leads the move to objective metrics, milestones, and Lean-Agile budgeting
- Establishes and maintains the systems and reporting capabilities
- Offer guidance for OKRs and KPIs
- Communicates and amplifies the portfolio's strategy
- Fosters more Agile contracts and leaner Supplier and Customer partnerships

Accelerating Flow

While flow-based guidance is embedded throughput SAFe, a five-article series directly addresses impediments to flow: 1. [Principle #6- Make value flow without interruptions](#), 2. [Portfolio Flow](#), 3. [Solution Train Flow](#), 4. [ART Flow](#), and 5. [Team Flow](#). These articles define flow with a set of eight flow accelerators that foster operational excellence. The LACE can coach RTEs, and Scrum Masters/Team Coaches to address, optimize, and debug issues with achieving continuous flow. The VMO has a primary responsibility in improving portfolio flow.

Lean Governance

Lean governance is the function that supports oversight of spending, audit, compliance, expenditure, measurement, and reporting. The Lean governance collaboration and responsibilities (Figure 6) require the active engagement of the VMO, LACE, Business Owners, and Enterprise Architects. The following sections

describe their duties.

Figure 6. Lean governance collaboration and responsibilities

Figure 6. Lean governance collaboration and responsibilities

Forecast and Budget Dynamically

As described earlier, SAFe provides a Lean approach to budgeting—*a* lightweight, more fluid, Agile process that replaces the fixed, long-range budget cycles, financial commitments, and fixed-scope expectations of a traditional planning process. This new approach to planning and budgeting includes understanding the historical and forecasted future costs associated with each Solution and forecasted epics costs. LPM adjusts budgets on a cadence, typically every six months or when significant events warrant, as part of the strategic portfolio review or [Participatory Budgeting](#) events (see below).

Measure Portfolio Performance

Each portfolio establishes the minimum metrics needed to measure portfolio performance to ensure:

- Progress with strategy implementation
- Alignment of strategy and execution
- Spending aligns with the agreed boundaries
- Business outcomes are continually improving without too much oversight of feature implementation
- LPM competency is improving

Measure and Grow is how portfolios evaluate their progress toward business agility and determine their next improvement steps. It consists of the following three measurement domains:

1. **Outcomes:** How well do the portfolio's solutions meet customers' needs and provide the expected results for the business?
2. **Flow:** How efficient is the portfolio at delivering a continuous flow of value to its customers and the desired outcomes for the business?
3. **Competency:** The LPM competency self-assessment enables organizations to evaluate their proficiency against the three dimensions of Strategy & Investment Funding, Agile Portfolio Operations, and Lean Governance.

The following sections describe these three domains.

Outcomes

A SAFe Portfolio primarily measures business outcomes by defining [Objectives and Key Results](#) (OKR) for Strategic Themes *and* Value Stream Key Performance Indicators (KPIs).

OKRs are a goal-setting framework that provides objective evidence of progress (Key Results) toward achieving a set of business outcomes (objectives). They help anchor ambitious goals with reality. OKRs facilitate breaking out of status quo thinking and enable the portfolio to explore new, often unknown, territory. If the portfolio has a big dream—and inspiring strategic theme—OKRs will help measure progress toward achieving it.

KPIs are specific and quantifiable measures of business results for the value streams within that portfolio. Outcome metrics of this kind are typically context-specific and depend heavily on the organization, business model, and the nature of solutions delivered to the customer. For example, the customer conversion rate may be a meaningful metric for an eCommerce business but would be inapplicable to a microchip manufacturer. Some indicators, however, may be successfully applied across contexts, such as the Net Promoter Score. The Value Stream KPIs article provides guidance for defining appropriate KPIs, informed by the strategic themes for that particular SAFe Portfolio.

Flow

As noted earlier, SAFe's [Measure and Grow](#) guidance offers portfolios a way to assess and improve their ability to deliver innovative business solutions quickly. It includes six measures specific to flow: *distribution, velocity, time, load, efficiency, and predictability*. *Flow time, load, and distribution* are particularly relevant to the *portfolio* and are briefly described below.

Flow time measures the interval needed for all the steps in the portfolio workflow to be completed. It can also be helpful to measure specific parts of this flow. For example, from the time the epic is pulled into the "review" state until its hypothesis has been evaluated.

Flow load indicates how many epics are currently in the system by process state. Keeping a healthy, limited portfolio WIP is critical to enabling the fast flow of strategic value.

Flow distribution measures the amount of each type of work in the portfolio for a given time. A helpful view of portfolio flow distribution illustrates the trend of money allocation across investment horizons.

The [Portfolio Flow](#) article guides LPM on accelerating flow and providing continuous epics to achieve the portfolio's vision and enterprise business objectives.

Competency

Measuring the level of organizational competency for a SAFe portfolio is accomplished with the following assessments:

SAFe Business Agility assessment is designed for business and portfolio stakeholders to assess their overall progress in achieving true business agility.

LPM core competency assessment helps the LPM team and its stakeholders measure their proficiency against the three domains of Strategy and Investment Funding, Agile Portfolio Operations, and Lean Governance.

Each of these assessments follows a standard process pattern of running the self-assessment, analyzing the results, taking action, and celebrating the successes. See the "Measuring Competency" section of the Measure and Grow article to download these assessments.

Coordinate Continuous Compliance

Lean audit and compliance provide continuous adherence while minimizing overhead and supporting the ongoing flow of value. This compliance may address internal or external financial auditing constraints and industry legal or regulatory

guidelines. These obligations impose significant limits on solution development and operations. Traditional compliance procedures tend to defer these activities to the end of the project. That subjects the enterprise to the risk of late discovery, subsequent rework, and even compromising regulatory or legal exposure. Therefore, SAFe recommends a more continuous approach to coordinate ongoing compliance with relevant standards.

For more information, see the advanced topic article and whitepaper: [Achieving Regulatory and Industry Standards Compliance with SAFe](#).

Assuring Big Data Security and Compliance

Every enterprise uses data to improve its products, optimize operations, and better understand its customers and markets. SAFe addresses big data concerns at the portfolio level as it requires vision, investment, and governance within and across the value streams within the portfolio. Big Data governance manages the availability, usability, integrity, and security of the data in enterprise systems based on internal data standards and policies controlling data usage. Effective data governance ensures that data is consistent and trustworthy and doesn't get misused.[1]

Lean Portfolio Management Events

The effective operation of the LPM function relies on three significant events:

1. Strategic Portfolio Review
2. Portfolio Sync
3. Participatory Budgeting

Typically, these events are held on a cadence, as illustrated in Figure 7.

Figure 7. Typical rhythm for the three LPM events

Figure 7. Typical rhythm for the three LPM events

Strategic Portfolio Review

The *strategic portfolio review* event provides ongoing strategy, implementation, and budget alignment. This event focuses on achieving and advancing the portfolio vision. It's typically held on a quarterly cadence, at least one month before the next PI Planning event, to enable value streams to prepare and respond to any changes,

Portfolio Sync

The *portfolio sync* provides visibility into how well the portfolio is progressing toward meeting its objectives. This event has a more *operational* focus than the strategic portfolio review. Topics typically include reviewing epic implementation, the status of KPIs, addressing dependencies, and removing impediments. The portfolio sync is generally held monthly and may be replaced with the strategic portfolio review on a given month.

Figure 8 compares the strategic portfolio review and portfolio sync events.

Figure 8. LPM events purpose and topics

Figure 8. LPM events purpose and topics

Participatory Budgeting

SAFe [Participatory Budgeting \(PB\)](#) is an LPM event in which a group of stakeholders decides how to invest the portfolio budget across solutions and epics. The resulting data is used to finalize adjustments to the value stream budgets. These budgets are typically adjusted twice annually using PB. If adjusted less frequently, spending is fixed for too long, limiting agility. Also, although more frequent budget changes may seem to support increased agility, they may create too much uncertainty and an inability to commit to any near-term course of action. (See Lean Budgets for more information).

Summary

Successfully defining and executing a strategy in a world of increasing uncertainty is

challenging. It requires modernizing portfolio management, applying Lean-Agile thinking, and organizing Agile teams and ARTs around value streams that deliver a continuous flow of value to the enterprise's customers.

Strategy and investment funding ensures the right work is happening at the right time. Continuous and early feedback on current initiatives, coupled with a Lean approach to funding, allows the portfolio to make the necessary adjustments to meet its business targets. Agile portfolio operations facilitate coordination across the portfolio's value streams, maintaining alignment between strategy and execution and fostering continued operational excellence. Lean governance closes the loop by measuring portfolio performance and supporting dynamic budget adjustments to maximize value. Collectively, these competencies work together to create superior economic outcomes.

Learn More

[1] <https://www.techtarget.com/searchdatamanagement/definition/data-governance>

Last update: 3 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite

[presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[Permissions Form](#)

[Usage and Permissions](#)

300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Kaizen is about changing the way things are. If you assume that things are all right the way they are, you can't do kaizen. So change something!

—Taiichi Ohno

Inspect and Adapt

[Adjust Cookie Settings](#)

The Inspect and Adapt (I&A) is a

Inspect & Adapt: Overview

significant event held at the end of each PI, where the current state of the Solution is demonstrated and evaluated. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop.

The Agile Manifesto emphasizes the importance of continuous improvement through the following principle: ÔAt regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.Ó

In addition, SAFe includes Ôrelentless improvementÕ as one of the four SAFe [Core Values](#) as well as a dimension of the [Continuous Learning Culture](#) core competency. While opportunities to improve can and should occur continuously throughout the PI (e.g., [Iteration Retrospectives](#)), applying some structure, cadence, and synchronization helps ensure that there is also time set aside to identify improvements across multiple teams and [Agile Release Trains](#).

Details

All ART stakeholders participate along with the [Agile Teams](#) in the I&A event. The result is a set of improvement backlog items that go into the [ART Backlog](#) for the next PI Planning event. In this way, every ART improves every PI. A similar I&A event is held by [Solution Trains](#).

The I&A event consists of three parts:

1. PI System Demo
2. Quantitative and qualitative measurement
3. Retrospective and problem-solving workshop

Participants in the I&A should be, wherever possible, *all* the people involved in building the solution. For an ART, this includes:

The Agile teams
[Release Train Engineer \(RTE\)](#)
[System and Solution Architects](#)
[Product Management, Business Owners](#), and other stakeholders

Additionally, [Solution Train](#) stakeholders may also attend this event.

PI System Demo

The PI [System Demo](#) is the first part of the I&A, and it's a little different from the regular system demos after every iteration. This demo shows all the [Features](#) the ART has developed during the PI. Typically the audience is broader; for example, [Customers](#) or [Portfolio](#) representatives are more likely to attend this demo.

Therefore, the PI system demo tends to be a little more formal, and extra preparation and setup are usually required. But like any other system demo, it should be timeboxed to an hour or less, with the level of abstraction high enough to keep stakeholders actively engaged and providing feedback.

Before or as part of the PI system demo, [Business Owners](#) collaborate with each Agile Team to score the actual business value achieved for each of their [Team PI Objectives](#), as illustrated in Figure 1.

The *achievement score* is calculated by separately totaling the business value for the *plan* and *actual* columns. The uncommitted objectives are not included in the total plan. However, they are part of the total actual. Then divide the planned total by the actual total to calculate the achievement score illustrated in Figure 1.

Figure 1. Scoring actual business value for each team

Figure 1. Scoring actual business value for each team

Quantitative and Qualitative Measurement

In the second part of the I&A event, teams collectively review any quantitative and qualitative metrics they have agreed to collect, then discuss the data and trends. In preparation for this, the RTE and the [Solution Train Engineer](#) are often responsible for gathering the information, analyzing it to identify potential issues, and facilitating the presentation of the findings to the ART.

Each team's planned vs. actual business value is rolled up to create the ART predictability measure, as shown in Figure 2.

Figure 2. ART predictability measure is rolled up from each team's planned vs. actual business value

Reliable trains should operate in the 80-100 percent range; this allows the business and its external stakeholders to plan effectively. (Note: Uncommitted objectives are excluded from the planned commitment. However, they are included

in the actual business value achievement, as can also be seen in Figure 1.)

Retrospective

The teams then run a brief (30 minutes or less) retrospective to identify a few significant issues they would like to address during the problem-solving workshop. There is no one way to do this; several different Agile retrospective formats can be used [3].

Based on the retrospective and the nature of the problems identified, the facilitator helps the group decide which issues they want to tackle. Each team may work on a problem, or, more typically, new groups are formed from individuals across different teams who wish to work on the same issue. This self-selection helps provide cross-functional and differing views of the problem and brings together those impacted and those best motivated to address the issue.

Key ART stakeholders—*including Business Owners, customers, and management*—join the retrospective and problem-solving workshop teams. The Business Owners can often unblock the impediments outside the team’s control.

Problem-Solving Workshop

The ART holds a structured, root-cause problem-solving workshop to address systemic problems. Root cause analysis provides a set of problem-solving tools used to identify the actual causes of a problem rather than just fixing the symptoms. The RTE typically facilitates the session in a timebox of two hours or less.

Figure 3 illustrates the steps in the problem-solving workshop.

Figure 3. Problem-solving workshop format

Figure 3. Problem-solving workshop format

The following sections describe each step of the process.

Agree on the Problem(s) to Solve

American inventor Charles Kettering is credited with saying that “a problem well stated is a problem half solved.” At this point, the teams have self-selected the problem they want to address. But do they agree on the details of the problem, or is it more likely that they have differing perspectives? To this end, the teams should spend a few minutes clearly stating the problem, highlighting the “what,” “where,” “when,” and “impact” as concisely as possible. Figure 4 illustrates a well-written problem statement.

Figure 4. Example problem statement

Figure 4. Example problem statement

Perform Root Cause Analysis

Effective problem-solving tools include the fishbone diagram and the Ô5 Whys.Õ Also known as an [Ishikawa Diagram](#), a fishbone diagram is a visual tool to explore the causes of specific events or sources of variation in a process. Figure 5 illustrates the fishbone diagram with a summary of the previous problem statement written at the head of the Ôfish.Õ

Figure 5. Fishbone diagram with primary sources identified

Figure 5. Fishbone diagram with primary sources identified

For our problem-solving workshop, the main bones often start with the default categories of people, processes, tools, program, and environment. However, these categories should be adapted as appropriate.

Team members then brainstorm causes that they think contribute to solving the problem and group them into these categories. Once a potential cause is identified, its root cause is explored with the [5 Whys](#) technique. By asking ÔwhyÕ five times, the cause of the previous cause is uncovered and added to the diagram. The process stops once a suitable root cause has been identified, and the same process is then applied to the next cause.

Identify the Biggest Root Cause

Pareto Analysis, also known as the 80/20 rule, is used to narrow down the number of actions that produce the most significant overall effect. It uses the principle that 20 percent of the causes are responsible for 80 percent of the problem. ItÕs

beneficial when many possible courses of action compete for attention, which is almost always the case with complex, systemic issues.

Once all the possible causes-of-causes are identified, team members then cumulatively vote on the item they think is the most significant factor contributing to the original problem. They can do this by dot voting. For example, each person gets five votes to choose one or more causes they think are most problematic. The team then summarizes the votes in a Pareto chart, such as the example in Figure 6, which illustrates their collective consensus on the most significant root cause.

Figure 6. Pareto chart of probable causes

Figure 6. Pareto chart of probable causes

Restate the New Problem

The next step is to pick the cause with the most votes and restate it clearly as a problem. Restating it should take only a few minutes, as the teams clearly understand the root cause.

Brainstorm Solutions

At this point, the restated problem will start to imply some potential solutions. The team brainstorms as many possible corrective actions as possible within a fixed timebox (about 15Ð30 minutes). The rules of brainstorming apply here:

- Generate as many ideas as possible
- Do not allow criticism or debate
- Let the imagination soar
- Explore and combine ideas

Create Improvement Backlog Items

The team then cumulatively votes on up to three most viable solutions. These potential solutions are written as improvement [stories](#) and features, planned in the following PI Planning event. During that event, the RTE helps ensure that the relevant work needed to deliver the identified improvements is planned. This approach closes the loop, thus ensuring that action will be taken and that people and resources are dedicated as necessary to improve the current state.

Following this practice, problem-solving becomes routine and systematic, and team members and ART stakeholders can ensure that the train is solidly on its journey of relentless improvement.

Inspect and Adapt for Solution Trains

The above describes a rigorous approach to problem-solving in the context of a single ART. If the ART is part of a Solution Train, the I&A event will often include key stakeholders from the Solution Train. In larger value streams, however, an additional Solution Train I&A event may be required, following the same format.

Due to the number of people in a Solution Train, attendees at the large solution I&A event cannot include everyone, so stakeholders are selected that are best suited to address the problems. This subset of people consists of the Solution Train’s primary stakeholders and representatives from the various ARTs and [Suppliers](#).

Learn More

- [1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.
- [2] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.
- [3] Derby, Esther, and Diana Larsen. *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf, 2009.

Last update: 22 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
 Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The most important single thing is to focus obsessively on the customer. Our goal is to be earth's most customer-centric company.

Jeff Bezos

Customer Centricity

Adjust Coo
Customers are the ultimate beneficiaries of the value of the solutions created and maintained by a portfolio's value streams.

Customer Centricity is a mindset that focuses on creating positive experiences for the customer through the full set of products and services that the enterprise offers.

Customer-centric organizations deliver whole-product solutions designed with a deep understanding of customer needs. This results in greater profits, increased employee engagement, and more satisfied customers in the private sector. Nonprofits and the public sector (governments) can achieve the resiliency, sustainability, and alignment needed to fulfill their mission.

Note: This article describes the *mindset* and impact of customer centricity. The related [Design Thinking](#) article provides the *tools and practices* to support creating desirable products that are profitable and sustainable over their lifecycle. Therefore, it's recommended that these two articles are read together.

Details

Customer centricity is a mindset that helps organizations make decisions that are based on a deep understanding of its effect on customers and end-users that motivates the following behaviors:

Focusing on the customer ▷ aligning and focusing the organization on specific, targeted user segments

Understanding the customer's needs ▷ moving beyond merely listening to customers who ask for features and investing the time to identify the customer's fundamental and ongoing needs

Thinking and feeling like the customer ▷ striving to see the world from their customer's point of view

Building whole product solutions ▷ designing a complete solution for the user's needs, and ensuring that the initial and long-term customer experience is continually evolving toward the ideal solution

Knowing customer lifetime value ▷ moving beyond a transactional mentality, and focusing on creating longer-term relationships based on a clear understanding of how the customer gains value

Driving Research

The foundation of the customer-centric enterprise is market and user research that creates actionable insights into the problems customers face, the [Solution](#)

[Context](#), and requirements. Market research helps drive *strategy*, while user research drives *design*, as shown in Figure 1.

Figure 1. Market and user research explore different aspects of the problem and solution space

Ongoing research activities are supported through [Continuous Exploration](#), automated data collection, and the feedback loops between the solution and its solution context.

Designing with Empathy

Designing with empathy puts aside preconceived ideas and helps create solutions from the customers' perspective. [1] It motivates teams to understand and experience the world from the customer's viewpoint, learning and appreciating the difficulties they face, their roles, and their context. It emphasizes user research,

including activities such as Gemba walks (for example, going to the customer's workplace). Gemba helps [Agile Teams](#) build empathy better to understand the user's emotional and physical needs. The way they see, understand and interact with the world around them.

Customer-centric enterprises apply *empathic design* throughout the product lifecycle, guiding the development of solutions that move beyond functional needs and address:

Aesthetic and emotional needs

Ergonomic requirements, such as the placement of physical features

Product attributes that users may not explicitly request, such as performance, security, and compliance, but which are essential for viability

An understanding of how the solution may impact the solution context

The impact of the solution on related or affected groups

The solution's architecture ensures operations, maintenance, and support of customer needs

Understanding Customer Engagement

Market research helps determine the nature of customer relationships, which is largely determined by the type of solution:

General solutions ▷ intended for a broad segment of customers

Custom-built solutions ▷ built and designed for a specific customer

Figure 2 illustrates the level of indirect or direct customer engagement in each case

Figure 2. Customer engagement models in general and custom-built solutions

Figure 2. Customer engagement models in general and custom-built solutions

General Solutions

General solutions must address the needs of a broader market or segment in which no single customer adequately represents the whole market. In this case, Product and Solution Management become the indirect customer proxy; they have authority over solution content. It's their responsibility to facilitate external interaction and ensure teams will hear the "voice of the customer" and that the organization will continuously validate new ideas. Scope, schedule, and budget for development are generally at the discretion of the internal Business Owners.

Since customers are unlikely to regularly participate in planning and [System Demo](#) events, their interaction is often based on requirements workshops, focus groups, usability testing, and limited beta releases. The solution evolves through feedback from user behavior analysis, metrics, and business intelligence to validate various hypotheses.

Custom-Built Solutions

External customers collaborate with Product and Solution Management for custom-built solutions in joint design efforts. While the customer leads the effort, deliverables, sequencing, and timing, need negotiation. This collaboration promotes incremental learning and creates opportunities to adjust plans based on the best available data.

SAFe's focus on cadence-based development directly supports the collaborations that create the best outcomes in custom-built solutions. For example, PI Planning provides the time and space to align all stakeholders around the next set of deliverables. The successful completion of [PI Objectives](#) establishes a high degree of trust in the collaborative development process and generates data that improves forecasting and economic modeling.

Deep and Narrow Solutions

Deep and narrow solutions are the middle ground between general and custom solutions. These solutions have a small number of customers who will often pay a significant amount of money for these products and services. For example, a solution to manage logistics for NFL stadiums with more than 50,000 people will serve a potential market of fewer than 32 customers. While maintaining the discipline of creating a single solution that answers a target market's needs, [Product](#) and [Solution](#) Management must leverage their familiarity with the small number of customers they're serving.

Multi-Segment Solutions

Some solutions serve different markets, each using the solution slightly differently. In this situation, customer-centricity requires understanding the unique needs of each segment. The following examples highlight the need for multi-segment solutions:

B2C □ A B2C software company serving hundreds of thousands to millions of indirect customers via a website may also offer a set of developer APIs to partners

B2B □ Members of a B2B partner segment may act more like customers of custom-built solutions, each making specific requests of the software provider to adjust, extend, or improve the API to meet their unique needs better

Whole Product Thinking

Just as [Systems Thinking](#) takes a holistic approach to solution development, *whole product thinking* involves viewing a product as more than just a sum of its features, but rather as everything involved with the experience customers have

with the purchase, use, and support of the product.

Companies must deliver the core product at a minimum. However, customers will not be delighted if it's missing critical aspects of the ownership experience. For example, suppose you buy a smartphone that doesn't have a headphone jack (or Bluetooth), or you must buy a separate charger, battery, or SIM card to operate the phone, customers will not be happy. Similarly, your customers will be dissatisfied if you have an excellent phone with no warranty or customer support or if it doesn't connect to wifi. You can have the most amazing product in the world, but if you don't pay attention to the Whole Product Concept, it may fail. [1]

Developing a complete product requires starting with an understanding of the real benefits that customers can expect to receive or experience from the product. After that, consider all aspects of the customer's experience, from purchase, first use, ongoing user experience, maintenance, add-ons, and accessories— even to how the product may be upgraded and supported. Understand all the touchpoints the customer will have with the product, such as setting up the phone with the preferred cellular carrier or the home wifi.

Customers are only satisfied if the product's actual value is the same or *exceeds* their perceived value. Kotler, Levitt, and Moore devised variations of a whole product model that recognizes the levels of customer needs. Figure 3 provides a version of this model that has been adapted for SAFe, followed by a brief description of each level.

Figure 3. Whole Product Model [6]

Figure 3. Whole Product Model [6]

1. **MVP** Ø is an early and minimal version of a new product used to prove or disprove the benefit hypothesis. The MVP helps answer the question: do customers and users want the solution, and can it be built? The SAFe MVP is an actual product that real customers can use and allows the enterprise to generate validated learning to determine the core product features and beyond. As indicated in Figure 3. the MVP is not a whole product but rather a starting point that proves the hypothesis.
2. **Core product** Ø addresses the basic functional needs of the customer. It Øs adequate to accomplish the jobs to be done minimally but lacks certain features or attributes of a product that the customer would expect.
3. **Expected product** Ø provides the attributes buyers usually expect and agree to when purchasing a product. The expected product includes the core product and other benefits the customer expects when purchasing it. For example, the product has online help, documentation, customer support during the warranty period, and so on.
4. **Augmented product** Ø provides the additional features, benefits, attributes, or related services that differentiate the product from its competitors to delight customers. For example, a laptop comes with free third-party add-ons, such as a password manager, VPN, touch screen, and more.
5. **Potential product** Ø envisions the features and other attributes necessary to attract and retain customers indefinitely. Informed by market and user research, the

potential product fuels longer-term strategic planning and creates opportunities for sustainable product advantages.

Leveraging Market Rhythms and Events

The [Lean-Agile Mindset](#) drives the continuous and sustainable flow of value to customers, motivating organizations to understand how the timing of specific releases influences their perceived value. In other words, the **value** of a product to customers and the organization can vary significantly based on the timing of its release. To create the highest value for all stakeholders, customer-centric organizations leverage market rhythms and market events: [3]

A *market rhythm* is a set of events *repeatedly occurring* on a *predictable cadence*. For example, retailers routinely prepare for the holiday shopping season by upgrading their systems to gain a competitive edge to support significantly higher transaction volumes.

A *market event* is a *one-time* future event with a high probability of materially affecting one or more solutions. They can be external, such as the launch of government regulations, or internally created, such as a company's annual user conference.

Market Rhythms

Market rhythms help companies recognize and capitalize on opportunities that are predictable and require longer-term planning. Figure 4 illustrates an example of the market rhythms of three different companies.

1. A **B2B retail software company** that offers real-time pricing updates must issue important alerts well before the shopping season. It must also update every point of sale terminal in 400 different stores and train all employees on the software's new capabilities.
2. A **B2C social media** company where the value over time is relatively constant suggests it is less affected by market rhythms [3].
3. A **toy maker** must deliver a Ôhot new toyÕ in time for the holiday shopping season, or its price and value will drop significantly!

Figure 4. An example of market rhythms for three different types of companies

Figure 4. An example of market rhythms for three different types of companies

Market Events

With an understanding of market rhythms, customer-centric road-mapping activities typically focus on the impact of market events. Figure 5 illustrates three market events, highlighted by the diamond-shaped milestones:

1. Known changes to regulations
2. Expected release by a competitor
3. Potential technology change

Figure 5. An example of market events

Figure 5. An example of market events

Market events are typically represented as milestones and strongly impact the timing for releasing solutions. They may also inform the content and timing of features or solution development activities identified during PI planning.

Understanding the Solution Context

Insights gained from the Gemba walks and other research activities define the functional and operational requirements of the solution's operating environment. In SAFe, this is known as the [Solution Context](#), which captures environmental, installation, operation, and support needs.

Understanding the solution context is critical to value delivery. It identifies constraints outside the organization's control. For example, a self-driving vehicle must drive and navigate icy roads while complying with motorist regulations. In another situation, the solution context may describe negotiated constraints, such

as when the organization uses principles of set-based design and collaborates with one or more [Suppliers](#) to optimize the total system's space, power requirements, and weight.

Accordingly, some aspects of the Solution Context are fixed, and some are negotiable; this creates a level of coupling between the Solution, Suppliers, and the Solution Context. The mandate of Business Agility motivates Product and Solution Managers to seek optimal solutions, including changing the Solution Context to encourage innovation.

Understanding Customer Value

Creating viable and sustainable offerings requires a deep understanding of the customer's perception of value. Consider a for-profit enterprise that has identified a customer problem that will cost 800K. If the customer perceives less than 800K of value from the solution, the organization will be unable to sell it at a price that creates a viable offering. And even if the customer perceives more than 800K of value, suggesting the enterprise can make a profit, the solution may not be sustainable if the revenue is insufficient to fund new and ongoing work.

Figure 6 illustrates two primary ways customers derive value from products and solutions:

Figure 6. Elements of customer value

Figure 6. Elements of customer value

There are several other aspects of value—*for example, brand value and aligning the organization’s values and beliefs with customers [3]*.

Learn More

[1] Leonard, Dorothy, and Jeffrey F. Rayport. *Spark Innovation Through Empathic Design*. Harvard Business Review, December 1997.

[2] Moore, Geoffrey. *Escape Velocity: Free Your Company’s Future from the Pull of the Past*. Harper Business, 2011.

[3] Hohmann, Luke. *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Addison-Wesley Professional, 2003.

[4] Levitt, Theodore. *Marketing Success Through Differentiation—of Anything*. Harvard Business Review, January 1980.

[5] <https://expertprogrammanagement.com/2017/10/five-product-levels/>

[6] Kotler, Philip, and Kevin Keller. *Marketing Management* 15th Edition.

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

[Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



*Stories act as a
Öpidgin language,Õ
where both sides
(users and
developers) can
agree enough to
work together
effectively.*

ÑBill Wake, co-
inventor of Extreme
Programming

Adjust Cookie Settings

Story

Stories are short descriptions of a small piece of desired functionality written from the user's perspective.

Agile Teams implement stories as small, vertical slices of system functionality that can be completed in a few days or less.

Stories are the primary artifact used to define system behavior in Agile. They are short, simple descriptions of functionality told from the user's perspective and written in their language. Each implements a small, vertical slice of system behavior.

Stories provide just enough information for business and technical people to understand the intent. Details are deferred until the story is ready to be implemented. Through acceptance criteria and acceptance tests, stories get more specific, helping to ensure system quality.

User stories deliver functionality directly to the end user. [Enabler](#) stories bring visibility to the work items needed to support exploration, architecture, infrastructure, and compliance.

Details

SAFe describes a four-tier hierarchy of artifacts that outline functional system behavior: [Epic](#), [Capability](#), [Feature](#), and Story. Collectively, these artifacts are used to describe the solution's intended behavior. The detailed implementation work is expressed through stories, which comprise the [Team Backlog](#). Some stories emerge from business and enabler features in the [ART Backlog](#), while others come from the team's local context.

Each story is a small, independent behavior that can be implemented incrementally and provides some value to the user or the [Solution](#). It's a vertical slice of functionality to ensure that every [Iteration](#) delivers new value. Stories are small and must be completed in a single iteration (see the splitting stories section).

Often, stories are first written on an index card or sticky note. The physical nature of the card creates a tangible relationship between the team, the story, and the user: it helps engage the entire team in story writing. Sticky notes also offer other benefits: they help visualize work and can be readily placed on a wall or table, rearranged in sequence, and even passed off when necessary. Stories allow an improved understanding of the scope and progress:

ÒWow, look at all these stories we are about to sign up forÓ (scope)

ÒLook at all the stories we accomplished in this iterationÓ (progress)

While anyone can write stories, approving them into the team backlog and accepting them into the system baseline are the Product Owner's responsibility. Of course, stickies don't scale well across the [Enterprise](#), so stories often move quickly into Agile Lifecycle Management (ALM) tooling.

There are two types of stories in SAFe, user stories and enabler stories, as described below.

Sources of Stories

Stories are typically driven by splitting business and enabler features, as Figure 1 illustrates.

Figure 1. Example of a business feature split into stories

Figure 1. Example of a business feature split into stories

User Stories

User stories are the primary means of expressing needed functionality. They essentially replace the traditional requirements specification. In some cases, however, they serve as a means to explain and develop system behavior later recorded in specifications supporting compliance, suppliers, traceability, or other needs.

Because they focus on the user as the subject of interest and not the system, user stories are value and customer-centric. To support this, the recommended form of expression is the “user-voice” form, as follows:

As a (user role), I want to (activity) so that (business value)

By using this format, the teams are guided to understand who is using the system, what they are doing with it, and why they are doing it. Applying the “user voice” format routinely tends to increase the team’s domain competence; they come to better understand the real business needs of their user. Figure 2 provides an example.

Figure 2. Example user story in user voice form

As described in [Design Thinking](#), personas describe specific characteristics of representative users that help teams better understand their end user. Example personas for the rider in Figure 2 could be a thrill-seeker “Jane” and a timid rider “Bob.” Stories descriptions can then reference these personas (As Jane I want).

While the user story voice is typical, not every system interacts with an end user. Sometimes the “user” is a device (for example, printer) or a system (for example, transaction server). In these cases, the story can take on the form illustrated in

Figure 3.

Figure 3. Example of a user story with a ŒsystemŒ as a user

Enabler Stories

Teams also develop the new architecture and infrastructure needed to implement new user stories. In this case, the story may not directly touch any end user. Teams use Œenabler storiesŒ to support exploration, architecture, or infrastructure. Enabler stories can be expressed in technical rather than user-centric language, as Figure 4 illustrates.

Figure 4. Example enabler story

Figure 4. Example enabler story

There are many other types of Enabler stories, including:

- Refactoring and **Spikes** (as traditionally defined in XP)
- Building or improving development/deployment infrastructure
- Running jobs that require human interaction (for example, indexing 1 million web pages)
- Creating the required product or component configurations for different purposes
- Verification of system qualities (for example, performance and vulnerability testing)

Enabler stories are demonstrated just like user stories, typically by showing the knowledge gained, artifacts produced, or the user interface, stub, or mock-up.

Writing Good Stories

Good stories require multiple perspectives. In Agile, the entire team creates a shared understanding of what to build to reduce rework and increase throughput. Teams collaborate using **Behavior-Driven Development** (BDD) to define detailed acceptance tests that definitively describe each story.

Collaborative story writing ensures all perspectives are addressed, and everyone agrees on the story's behavior with the results represented in the story's description, acceptance criteria, and acceptance tests. The acceptance tests are written using the system's domain language using BDD. BDD tests are then automated and run continuously to maintain **Built-In Quality**. The BDD tests are written against system requirements (stories) and, therefore, can be used as the definitive statement for the system's behavior, replacing document-based specifications.

The 3Cs: Card, Conversation, Confirmation

Ron Jeffries, one of the inventors of XP, is credited with describing the 3Cs of a story:

Card ☐ Captures the user story's statement of intent using an index card, sticky note, or tool. Index cards provide a physical relationship between the team and the story. The card size physically limits story length and premature suggestions for the specificity of system behavior. Cards also help the team "feel" upcoming scope, as there is something materially different about holding ten cards in one's hand versus looking at ten lines on a spreadsheet.

Conversation ☐ Represents a "promise for a conversation" about the story between the team, customer (or the customer's proxy), the PO (who may be representing the customer), and other stakeholders. The discussion is necessary to determine the more detailed behavior required to implement the intent. The conversation may spawn additional specificity in the form of acceptance criteria (the confirmation below) or attachments to the user story. The conversation spans all

steps in the story's life cycle:

- Backlog refinement
- Planning
- Implementation
- Demo

These just-in-time discussions create a shared understanding of the scope that formal documentation cannot provide. Specification by example replaces detailed documentation. Conversations also help uncover gaps in user scenarios and NFRs.

Confirmation ▷ The acceptance criteria provide the information needed to ensure that the story is implemented correctly and covers the relevant functional and NFRs. Figure 5 provides an example. Some teams often use the confirmation section of the story card to write down what they will demo.

Figure 5. Story acceptance criteria with BDD

Figure 5. Story acceptance criteria with BDD

Agile Teams automate acceptance tests wherever possible, often in business-

readable, domain-specific language. Automation creates an executable specification to validate and verify the solution. Automation also provides the ability to quickly regression-test the system, enhancing [Continuous Integration](#), refactoring, and maintenance.

Investing in Good Stories

Agile teams spend significant time discovering, elaborating, and understanding user stories and writing acceptance tests. This is as it should be, because it represents the fact that:

Writing the code for an understood objective is not necessarily the most challenging part of software development.

Instead, it is understanding the real objective of the code. Therefore, investing in good user stories, albeit at the last responsible moment, is a worthy effort for the team. Bill Wake coined the acronym INVEST [1] to describe the attributes of a good user story.

- I** Ø Independent (among other stories)
- N** Ø Negotiable (a flexible statement of intent, not a contract)
- V** Ø Valuable (providing a valuable vertical slice to the customer)
- E** Ø Estimable (small and negotiable)
- S** Ø Small (fits within an iteration)
- T** Ø Testable (understood enough to know how to test it)

Splitting Good Stories

Smaller stories allow faster, more reliable implementation since small items flow through any system faster, with less variability and reduced risk. Therefore, splitting bigger stories into smaller ones is a mandatory skill for every Agile team. It's both the art and the science of incremental development. *Agile Software Requirements* describes ten ways to split stories [1]. A summary of these techniques follows:

- Workflow steps
- Business rule variations
- Major effort
- Simple/complex
- Variations in data
- Data entry methods
- Deferred system qualities
- Operations (ex., Create, Read, Update, Delete [CRUD])
- Use-case scenarios
- Break-out spike

Figure 6 illustrates an example of splitting by use-case scenarios.

Figure 6. An example of splitting a big Story into smaller stories

Figure 6. An example of splitting a big Story into smaller stories

Estimating Stories

Agile teams use story points and “estimating poker” to value their work [1, 2]. A story point is a singular number that represents a combination of qualities:

- Volume** ☰ How much is there?
- Complexity** ☰ How hard is it?
- Knowledge** ☰ What’s known?
- Uncertainty** ☰ What’s unknown?

Story points are relative, without a connection to any specific unit of measure. Each story’s size (effort) is estimated relative to the smallest story, which is assigned a size of “one.” A modified Fibonacci sequence (1, 2, 3, 5, 8, 13, 20, 40, 100) is applied that reflects the inherent uncertainty in estimating, especially large

numbers (for example, 20, 40, 100) [2].

Estimating Poker

Agile teams often use *Estimating poker*, which combines expert opinion, analogy, and disaggregation to create quick but reliable estimates. Disaggregation refers to splitting a story or feature into smaller, easier-to-estimate pieces.

(Note that there are several other methods used as well.) The rules of estimating poker are:

- Participants include all team members.
- Each estimator is given a deck of cards with 1, 2, 3, 5, 8, 13, 20, 40, 100, *, and ?
- The PO participates but does not estimate.
- The *Scrum Master/Team Coach* participates but does not estimate unless they are doing actual development work.
- For each backlog item to be estimated, the PO reads the story's description.
- Questions are asked and answered.
- Each estimator privately selects an estimating card representing their estimate.
- All cards are turned over at the same time to avoid bias and to make all estimates visible.
- High and low estimators explain their estimates.
- After a discussion, each estimator re-estimates by selecting a card.
- The estimates will likely converge. If not, the process is repeated.

Some amount of preliminary design discussion is appropriate. However, spending too much time on design discussions is often a wasted effort. The real value of estimating poker is agreeing on a story's scope. It's also fun!

Velocity

The team's velocity for an iteration is equal to the sum of the points for all the completed stories that met their definition of done (DoD). As the team works together over time, their average velocity (completed story points per iteration) becomes reliable and predictable. Predictable velocity assists with planning and helps limit Work in Process (WIP), as teams don't take on more stories than their historical velocity would allow. This measure also estimates how long it takes to deliver epics, features, capabilities, and enablers, which are also forecasted using story points.

Capacity

Capacity is the portion of the team's velocity that is available for any given iteration. Vacations, training, and other events can make team members

unavailable to contribute to an iteration's goals for some portion of the iteration. This decreases the maximum potential velocity for that team for that iteration. For example, a team that averages 40 points delivered per iteration would adjust their maximum velocity down to 36 if a team member is on vacation for one week. Knowing this in advance, the team only commits to a maximum of 36 story points during iteration planning. This also helps during PI Planning to forecast the actual available capacity for each iteration in the PI, so the team doesn't over-commit when building their PI Objectives.

Starting Baseline for Estimation

In standard Scrum, each team's story point estimating¹ and the resulting velocity² is a local and independent concern. At scale, it becomes difficult to predict the story point size for larger epics and features when team velocities vary wildly. To overcome this, SAFe teams initially calibrate a starting story point baseline where one story point is defined roughly the same across all teams. There is no need to recalibrate team estimation or velocity. Calibration is performed one time when launching new [Agile Release Trains](#).

Normalized story points provide a method for getting to an agreed starting baseline for stories and velocity as follows:

- 1 Give every developer-tester on the team eight points for a two-week iteration (one point for each ideal workday, subtracting two days for general overhead).
- 2 Subtract one point for every team member's vacation day and holiday.
- 3 Find a small story that would take about a half-day to code and a half-day to test and validate. Call it a "one."
- 4 Estimate every other story relative to that "one."

Example: Assuming a six-person team composed of three developers, two testers, and one PO, with no vacations or holidays, then the estimated initial velocity = 5×8 points = 40 points/iteration. (Note: Adjusting slightly lower may be necessary if one of the developers and testers is also the Scrum Master/Team Coach.)

In this way, story points are somewhat comparable across teams. Management can better understand the cost for a story point and more accurately determine the cost of an upcoming feature or epic.

While teams will tend to increase their velocity over time³ and that's a good thing⁴ in reality, the number tends to remain stable. A team's velocity is far more affected by changing team size and technical context than by productivity variations.

Note: SAFe Team Kanban teams typically spend less time estimating stories than scrum teams do. In the Kanban flow-based model, work items or stories are typically split and sized so that the team can generally deliver a story within a few days. In the context of SAFe where teams need to participate in iteration planning and assign stories to future iterations, some notion of sizing is required.

SAFe Kanban teams may initially use estimating poker or a similar mechanism to size their stories. More likely, however, they develop a sense of breaking work into stories that are similar in size, as that assists flow in general and assures that no large story blocks other stories that also need to make their way through the Kanban system. As they understand their velocity, they are able to understand how many stories they can deliver in a unit of time, allowing them to place stories in iterations during PI Planning and to be able to make commitments to other teams as to when specific stories would be available.

For teams doing regular maintenance and support activities, estimating their normal backlog items often has less value. In many cases, these teams do not estimate this type of response work. However, all teams have retro items, potential improvements to their CD pipeline, and other significant tasks that require attention, scheduling, and estimating.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Cohn, Mike. *User Stories Applied: For Agile Software Development*. Addison-Wesley, 2004.

Last update: 7 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California](#)
- [Consumer Rights](#)

English

Scaled Agile Framework



None of my inventions came by accident. I see a worthwhile need to be met, and I make trial after trial until it comes.

—Thomas Edison

Iterations

Adjust Cookie Settings

Note: For more on SAFe Scrum, please read the additional Framework

articles in the Scrum series, including [SAFe Scrum](#), [SAFe Scrum Master/Team Coach](#), [Iteration Planning](#), [Iteration Goals](#), [Iteration Review](#), and [Iteration Retrospective](#)

Iterations are a standard, fixed-duration timebox during which Agile Teams and ARTs individually and collectively deliver incremental customer value while working towards the PI objectives.

Each [Planning Interval \(PI\)](#) typically has four two-week development Iterations (the subject of this article) followed by one [Innovation and Planning \(IP\)](#) Iteration. During these iterations, Agile Teams continuously explore, define, build, test, and deliver value to their customer. The IP iteration is an estimating buffer for meeting [PI Objectives](#) and provides dedicated time for innovation, continuing education, [PI Planning](#), and [Inspect and Adapt \(I&A\)](#) events.

[Agile Teams](#) on the [Agile Release Train](#) (ART) collaborate within a PI to advance the solution toward the team and ART PI Objectives. All teams on the ART synchronize to the same iteration and PI cadence.

Teams plan, demo, and learn together, which helps them focus on both local concerns and the larger aim of the train. This alignment also enables teams to explore, integrate, deploy, and release value together and independently.

Details

[Agile Teams](#) and ARTs fulfill their responsibilities by working in a series of iterations. Each is a Plan-Do-Check-Adjust (PDCA) for the ART. Iterations are continuous and sequential, and a new iteration starts immediately after the previous one.

The PDCA cycle has four steps as follows:

- Plan:** Determine the goals for the timebox and identify the work needed
- Do:** Rapidly deliver small increments of the Solution
- Check:** Review, demonstrate, and analyze the results and learnings
- Adjust:** Adjust and start a new cycle

As [Principle #7](#) [Apply cadence, synchronize with cross-domain planning](#) describes how Agile Teams can better synchronize with other groups and manage dependencies by following the same rhythm. However, the practices for planning and execution within that rhythm may differ based on whether the team

works in [SAFe Scrum](#) or [SAFe Team Kanban](#), or a hybrid of methods. Scrum teams use standard and specific iteration events (see the [SAFe Scrum](#) article) to plan and manage their work. Kanban teams typically work in a continuous flow model but may also use iteration events.

Figure 1. illustrates the nested PCDA cycles for the PI, ART, and Agile Teams.

Figure 2. Nested iteration PDCA cycles within the PI

Figure 1. Nested iteration PDCA cycles within the PI

The following sections describe each element of these PDCA cycles.

PI PDCA Cycle

PI Planning starts the PI PDCA cycle, where teams partially fill their backlogs to meet the solution vision. Subsequently, the ART and its teams execute individual and combined iteration PDCA cycles within the PI. The ART objectively measures progress with a regular series of [System Demos](#). The timebox

ends with the PI System Demo, the first part of the Inspect and Adapt (I&A) event. This demo shows all the [Features](#) the ART has developed during the entire PI. During the I&A, [Business Owners](#) score PI objectives and assess product/solution performance. Additionally, everyone in the ART collaborates to identify and address systemic problems in the problem-solving workshop portion of the I&A.

The following sections describe the ART and Agile Team activities within each iteration in the PI.

Agile Release Train Activities

Getting feedback is critical to an ART's high development velocity. Speed comes from fast learning, frequent integration points, and adaptation within the ART iteration PDCA cycles.

Figure 1. illustrates that each iteration is a short PDCA cycle in which the ART develops a significant portion of the PI objectives (See SAFe [Principle #4, Build incrementally with fast, integrated learning cycles](#), for more information.)

The following sections describe the activities for applying the PDCA cycle to the ART's iterations – the combined work of all its teams.

Plan

Planning for the system demo Ⓛ Teams and ART stakeholders plan on what will be presented at the system demo and who will facilitate and demonstrate. The [System Context](#), logistics, infrastructure, and staging environments must be established before the demo. [Product Management](#) communicates the system demo's agenda and content to stakeholders, including those with business and customer interest in the new functionality.

Planning for releases Ⓛ During the upcoming iteration, the teams will continuously integrate new functionality into the solution's baseline and [Release on Demand](#). The ART needs to plan for these releases and be ready to assess compliance and quality standards and business and customer impact.

Do

Delivering Value Ⓛ Completed features are released to customers to achieve the intended outcomes. While the ART can automate many release elements via the [Continuous Delivery Pipeline](#), manual oversight is often required, such as

communicating with ART stakeholders and engaging them in the delivery process.

Removing impediments and addressing risks ▷ ART stakeholders—particularly the [Release Train Engineer](#) (RTE), [Product Management](#), and [System Architect](#)—work with teams throughout the iteration to address obstacles. Occasionally, escalation to [Business Owners](#) and other stakeholders may be needed to remediate critical issues.

Executing the system demo ▷ The system demo provides an opportunity to inspect the ART’s combined increment at the end of every iteration, assess progress, get feedback, and improve both the product and development process.

Check

Conducting sync events ▷ The ART holds synchronization meetings (for example, PO and coach syncs) to ensure progress, help facilitate the resolution of cross-team and ART dependencies and resolve other problems.

Measuring and monitoring progress ▷ Progress towards PI objectives is measured and monitored. The ROAM board created during PI planning is typically reviewed during the PO or coach syncs to ensure those responsible for owning or mitigating a risk take the necessary actions.

Reviewing the progress of improvements from the previous I&A ▷ The RTE and [Scrum Master/Team Coaches](#) work together to ensure the ART completes the relevant work needed to accomplish the identified I&A improvements during the PI.

Reviewing outcome and flow metrics ▷ SAFe’s three measurement domains, *outcomes*, *flow*, and *competency*, are periodically reviewed to assess progress, support better decision-making, and help identify improvement opportunities for the ART’s people, products, and processes. (See the [Measure & Grow](#) article for more information.)

Adjust

Continually refine the ART Backlog ▷ The ART backlog is continuously prioritized and adjusted to ensure that progress is tracking to the PI objectives and that customers benefit from the solutions under development.

Improving the ART’s processes ▷ Measuring progress and gaining fast feedback provides the data the ART needs to improve the development process.

Addressing feedback ▷ Getting rapid feedback from [Customers](#), Business Owners, and other stakeholders is essential to ensure the ART is building a solution that is feasible, viable, desirable, and sustainable (see [Design Thinking](#)).

Team Activities

Agile Teams achieve a fast, reliable flow of value to the Customer by executing a full plan-do-check-adjust (PDCA) cycle each iteration. Activities include:

Plan

Refining the Team Backlog Teams refine their backlogs in a collaborative dialogue between the teams, Customers, and other stakeholders. This refinement breaks down barriers between the business and the development team, helps identify and address dependencies, and eliminates waste, handoffs, and delays. Story acceptance criteria increase the requirements' clarity, leverage the team's collective knowledge and creativity, and create understanding, engagement, and ownership.

Team iteration planning SAFe Scrum teams collaborate at the start of each iteration during the [Iteration Planning](#) event. They determine how much team backlog they can deliver during the upcoming timebox and record that work in the iteration backlog. They summarize their plan into a set of [Iteration Goals](#). SAFe Kanban teams also plan as needed (often weekly) to coordinate their work, replenish stories in their backlog, and address dependencies and fixed date commitments for the iteration. They often find it helpful to create iteration goals, which provide stakeholders and management with a shared language for maintaining alignment, communicating progress, managing dependencies, and making necessary adjustments during PI execution.

Planning for team and system demos Teams plan how to demo the stories they defined during and after PI planning to implement features and meet their PI objectives. Beginning with the end in mind facilitates planning and alignment, fostering a more thorough understanding of the functionality needed ahead of iteration execution.

Do

Delivering stories and high-value quality increments of value Teams create value within the ART's development cadence and synchronization requirements, releasing on demand as business needs dictate. They build and utilize a [Continuous Delivery Pipeline](#) (CDP), which helps ensure that [Continuous Exploration](#), [Continuous Integration](#), and [Continuous Deployment](#) activities occur as continually as possible.

Conducting team sync and demos Teams continuously coordinate and synchronize their work, inspecting progress and creating an action plan to coordinate current and upcoming work. They demo completed work immediately to get fast feedback.

Monitoring the progress of stories Teams use a Kanban board to visualize all active and pending work, workflow states, WIP limits, risks, and impediments.

Check

Conducting retrospectives ⚡ Teams reflect on the iteration and derive new ideas to improve the increment and process. Depending upon the method, teams may do this reflection at the [Iteration Retrospective](#) or periodically as needed. Retrospectives help ensure continuous improvements and that the knowledge gained can inform the problem-solving part of the I&A.

Reviewing improvement stories from prior iterations ⚡ Teams review the results of their previous improvement stories to ensure they have sufficiently addressed identified problems and their root causes.

Reviewing demos with the team’s stakeholders ⚡ Teams demo their completed work as soon as possible to get fast Customer and stakeholder feedback. Scrum teams typically conduct a more formal [Iteration Review](#) to understand their progress better.

Adjust

Refining the team backlog ⚡ Teams continually refine their backlog to ensure it always contains some stories ready for implementation. During refinement, teams review upcoming user stories and Features to discuss, estimate and establish preliminary acceptance criteria.

Improving the team’s processes ⚡ Based on the outcome, flow, and competency metrics, the team identifies their process improvements and addresses impediments and delays that otherwise interrupt the flow of value.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020

[2] Cockburn, Alistair. Using Both Incremental and Iterative Development. STSC CrossTalk 21, 2008.

Last update: 20 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



We are all familiar with guardrails on highways. They are put there to keep a simple mishap from turning into a full-blown catastrophe. If you go a little off course, the rails help you regain the path towards your destination.

~Anonymous

[Adjust Cookie Settings](#)

Lean Budget Guardrails

Lean Budget Guardrails describe the policies and practices for budgeting, spending, and governance for a specific portfolio.

SAFe provides Lean budgeting strategies that eliminate traditional project-based funding and cost accounting overhead. In this model, LPM maintains appropriate levels of oversight through allocating value stream budgets and applying Lean budget guardrails. This way, enterprises can have the best of both worlds: a development process far more responsive to market needs and professional and accountable spending management.

Details

Every SAFe portfolio operates within an approved budget for developing and deploying systems and Solutions that the Enterprise needs to meet its strategic objectives. As described in the [Lean Budgets](#) article, the portfolio's total budget is allocated to individual value streams by [Lean Portfolio Management](#) (LPM) and portfolio stakeholders. The value stream's budget funds the people and resources to help achieve the current [Portfolio Vision](#) and [Roadmap](#).

Establishing guardrails helps ensure that the mix of investments addresses both near-term opportunities and long-term strategy, that investments in technology, infrastructure, and maintenance aren't routinely ignored, and that significant investments are approved appropriately. Figure 1 illustrates four Lean budget guardrails:

1. Guiding investments by horizon
2. Applying capacity allocation to optimize value and solution integrity
3. Approving significant initiatives
4. Continuous Business Owner engagement

Figure 1. SAFe Lean budget guardrails

Figure 1. SAFe Lean budget guardrails

The first two guardrails are quantitative, guiding the allocation of investments within the approved budgets. The last two are process-related and are mainly qualitative, establishing how the budgets are governed. These guardrails are described in the sections that follow.

Guardrail 1: Guiding Investments by Horizon

As described in Lean Budgets, portfolio investments are organized by investment horizons that reflect four-time horizons. The amount of budget a given value stream allocates to solutions in these horizons determines the near- and long-term health of both the value streams and portfolio.

For example, a value stream solely focused on a Horizon 1 solution may be under-investing in future solution innovations, creating long-term risk. This may be balanced by the portfolio's intention to move the solution into Horizon 0 for

subsequent decommissioning to enable the value stream to focus on other, more promising solutions. Accordingly, LPM establishes portfolio-level guidance for investments to optimize the whole while promoting decentralization so that individual value streams can optimize their solutions, as Figure 2 illustrates.

Figure 2. Investment horizon budget guardrail

Figure 2. Investment horizon budget guardrail

Figure 2 shows that LPM has established different allocations for investments in solutions for each investment horizon. While this may be a healthy mix for a technology business, every portfolio and value stream has to consider its current context in determining its investment allocations for each horizon. A newly created value stream might allocate significantly more of its budget to Horizon 2 because it simply doesn't have any solutions in Horizon 1. For example, an established value stream retiring legacy solutions with substantial technical obsolescence might allocate more budget to Horizon 0. For more guidance on allocations for various industries, see [2].

Guardrail 2: Apply Capacity Allocation

Lean Budgeting enables decentralized decision-making and more efficient execution. However, every train is challenged with balancing the backlog of new business Features with investment in the Architectural Runway continuously—for example, maintaining current systems, avoiding velocity reduction, and the need for wholesale replacement of components or solutions due to obsolete technology.

Balancing business features and enablers complicates prioritizing work since different forces can pull the teams in different directions, as Figure 3 shows.

Figure 3. Balancing business features and enablers

Figure 3. Balancing business features and enablers

One solution to this challenge is that value streams (and ARTs) apply *capacity allocation* as a *quantitative* guardrail to determine how much of the total effort can be allocated for each type of activity for an upcoming PI, as shown in Figure 4.

Each value stream should adapt the capacity allocation categories or add new ones as needed.

Figure 4. Balancing the forces on the ART backlog

Figure 4. Balancing the forces on the ART backlog

Capacity Allocation Example Policies

Each value stream should develop explicit policies for managing capacity allocation. Following are *example* policy statements that many ARTs and Solution Trains have found helpful:

At each PI boundary, we agree on the percentage of capacity devoted to new features (or capabilities) versus enablers and tech debt and maintenance. Other capacity types may also apply, such as the percentage of capacity allocated to a specific epic.

We agree that **Product** and **Solution Management** have the authority to prioritize ART and Solution Train backlog items.

We agree to prioritize the business and enabler features and capabilities based on

economics and in collaboration with architects.

We agree to collaborate on sequencing work in a way that maximizes customer value and minimizes technical debt.

While the agreed-to policies can persist for some time, the amount of capacity allocated will change periodically based on the context. In an ART context, the capacity allocation decision can be revisited as part of backlog refinement in preparation for PI planning. At the same time, Solution Management and Solution Architects make similar choices ahead of pre-PI planning.

Guardrail 3: Approving Significant Initiatives

While each value stream is funded to promote empowerment and local decision-making authority, it is reasonable to ensure that significant investments are governed responsibly.

Figure 5. Epics that exceed the portfolio threshold require LPM approval

Figure 5. Epics that exceed the portfolio threshold require LPM approval

Figure 5. shows that a significant initiative has been identified. It then goes through a decision filter to determine whether or not it exceeds the portfolio epic threshold, which LPM establishes.

Below threshold: If the epic estimate is below the portfolio epic threshold, approval is managed through the Program or Solution Train Kanban systems.

Above threshold: If the epic estimate exceeds the portfolio epic threshold, it requires review and approval through the Portfolio Kanban system, regardless of which level the initiative originates. LPM defines the Portfolio Epic threshold to determine which Epics are a portfolio concern. Example thresholds include forecasted epic cost, forecasted number of PIs to implement epic, the strategic importance of the epic, or a combination of these factors.

Guardrail 4: Continuous Business Owner Engagement

Business Owners are uniquely qualified to ensure that the funding allocated to value streams is going toward the right things. Therefore, they serve as a critical guardrail that ensures that the priorities of the ARTs and Solution Trains are in alignment with LPM, customers, and Product and Solution Management, as illustrated in Figure 6.

Figure 6. Continuous Business Owner engagement

Figure 6. Continuous Business Owner engagement

Figure 6 shows the minimum activities Business Owners should actively participate in before, during, and after PI execution. They are briefly described next and covered in more detail in the [Business Owners](#) article.

Preparing for the upcoming PI During PI execution Business Owners ensure that ARTs and Solution Trains are allocating sufficient capacity for features, enablers, and technical debt and maintenance, as well as providing input on prioritization of Features and Capabilities using [Weighted Shortest Job First \(WSJF\)](#). Business Owners also collaborate with Product and Solution Management to ensure that the work planned for the PI contains the right mix of investments that address near-term opportunities (horizon 1) and long-term strategy (horizons 2 and 3) and that sufficient capacity is allocated for decommissioning solutions (horizon 0).

PI Planning During PI planning Business Owners actively participate in key activities, including the presentation of the vision, draft plan review, assigning business value to team PI objectives, and approving final plans.

Inspect & Adapt (I&A) Workshop During the [I&A workshop](#), Business Owners provide feedback on the solution's *fitness for purpose* during the [System Demo](#) (or [Solution Demo](#)). The Business Owner's feedback is critical, as only they can give the guidance the train needs to stay on course or take corrective action.

Additionally, they help assess actual value achieved versus plan and participate in the upcoming problem-solving workshop.

Learn More

[1] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, 2011.

[2] The Harvard Business Review article at <https://hbr.org/2012/05/managing-your-innovation-portfolio> provides example investment percentages for each horizon.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

Become a Trainer

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*Making and meeting
small commitments
builds trust.*

Nonaka and
Takeuchi, The
Knowledge-Creating
Company

PI Objectives

PI Objectives summarize the business and technical goals that teams and trains Adjust Coordinating to achieve in the upcoming PI.

During [PI Planning](#), teams create PI objectives they intend to accomplish in the upcoming [PI](#). These provide several benefits:

- Provide a common language for communicating with business and technology stakeholders
- Creates the near-term focus and vision
- Enables the ART to assess its performance and the business value achieved via the [ART Predictability Measure](#)
- Communicates and highlights each team's contribution to business value
- Expose dependencies that require coordination

Details

SAFe relies on a rolling wave of *short-term* commitments from [Agile teams](#) and trains to assist with business planning and outcomes, resulting in improved alignment and trust between development and business stakeholders. These are communicated via PI objectives.

While development is uncertain by its very nature, the business depends on teams for some amount of reliable, predictable forecasting. Too little, and the company can't plan. Too much, and the organization has committed to longer-term plans, which are unreliable at best and limit agility. Business and technology stakeholders need something in between, which is a primary purpose of PI objectives. In addition to alignment, setting realistic objectives also helps avoid too much work-in-process (WIP) in the system. PI objectives are built largely bottom-up as the teams identify them during PI planning.

Building the Team PI Objectives

During PI planning, the teams get presented with new [Features](#) and plan the [Stories](#) they need to deliver alongside stories representing work from their local context. This work is described as a set of specific team PI objectives. Doing so requires estimating and planning, knowledge of the team's capacity, analysis of upcoming features, defining stories for the [Team Backlog](#), and summarizing the information into simple business terms everyone can understand.

As for the number of objectives, a team should establish, there is no fixed rule, but 7-10 committed objectives (plus 2-3 uncommitted; see below) seem to work best. Beyond this threshold, other teams and business partners find the details challenging to understand and process. Plus, there are too many to review and process in a medium to large ART. Less, and the level of abstraction or aggregation is probably too high to be measured objectively at the end of the PI.

Figure 1 illustrates an example of one team's PI objectives.

Figure 1. A team's PI objectives

Figure 1. A team's PI objectives

Differentiate between Features and PI Objectives

The team's PI objectives often relate directly to intended features. Many are the same. However, the mapping is not always straightforward since some features require the collaboration of multiple teams, as Figure 2 illustrates.

Figure 2. From features to objectives, some features will appear in more than one team's PI objectives

Note that an individual team can deliver some features (such as Feature A); others (Feature B) require the collaboration of several teams. In addition to features and inputs to features, other team objectives will also appear. These can include technical objectives (for example, the proof of concept in Figure 1) that enable future features, enhancements to development infrastructure, milestones, and others. All the planning process results are captured in the team's objectives.

Features and acceptance criteria are excellent tools to help understand, capture, and collaborate around the work that needs to be done. Still, it's all too easy to get caught up in "finishing the features" and missing the overall goals hiding inside. PI objectives help shift focus away from developing features to achieving the desired business outcomes.

A better understanding of the intent offered by direct conversations with the **Business Owners** often results in the teams providing new perspectives to **System Architects** and **Product Management** and quickly finding ways to apply their expertise

to create better solutions.

(Note: The extended guidance article [Role of PI Objectives](#) further explains the differences between team PI objectives and features and provides additional insights into their usage and value.)

Committed and Uncommitted Objectives

Committing to and delivering a series of short-term objectives helps to build trust. Trust allows all stakeholders to move forward confidently and base decisions and plans on what is Ôvery likely to be true very soon.Ô But planning confidently in the face of the uncertainty inherent in research and development is difficult. Things donÕt always go as planned, and building some small amount of buffer into the system is prudent. If the buffer is too big, the ART might accomplish less than would otherwise be the case. If the buffer is too small, many commitments may not be feasible. As a result, planning and confidence erode. To address this, SAFe recommends teams use both committed and **uncommitted** objectives during planning. Uncommitted objectives help improve the predictability of delivering business value since they are *not* included in the teamÔs commitment or counted against teams in the ART predictability measure.

Uncommitted objectives are used to identify work that can be **variable** within the scope of a PI. The work is planned, but the outcome is not certain. Teams can apply uncommitted objectives whenever there is low confidence in meeting the objective. This low confidence can be due to many circumstances:

Dependencies with another team or supplier that cannot be guaranteed.

The team has little to no experience with functionality of this type. In this case, the teams may plan [Spikes](#) early in the PI to reduce uncertainty.

There are a large number of critical objectives that the business depends on, and the team is already loaded close to full capacity.

In this case, a few (no more than 2-3) uncommitted objectives are prudent. However, teams do their best to deliver the uncommitted objectives, and they are included in the capacity and plan for the PI. However, stakeholders plan accordingly since these objectives might not be finished in the PI.

Uncommitted objectives provide several benefits:

Improved economics Ô Without uncommitted objectives, a team commits to a 100 percent scope in a fixed timebox. This forces teams to trade off quality or build other buffers into the system. The other buffers can accumulate and convert Ôuncertain earliness to certain lateness,Ô resulting in less overall throughput.

Increased reliability ☐ Uncommitted objectives represent variable scope, allowing confidence in delivering the main priorities. In turn, delivering on the stated commitments is the most important factor in building trust between the teams and the stakeholders.

Adaptability to change ☐ To reliably deliver on a cadence, uncommitted objectives provide the capacity margin needed to meet commitments, yet alter priorities if necessary, when fact patterns change.

Write SMART PI Objectives

Team PI objectives summarize a team's plan for the PI. They are critically important. Sometimes the descriptions may be very technical and a little vague. As a countermeasure, teams make their objectives SMART:

Specific ☐ States the intended outcome as concisely and explicitly as possible. (Hint: Try starting with an action verb.)

Measurable ☐ It should be clear what a team needs to do to achieve the objective. The measures may be descriptive, yes/no, quantitative, or provide a range.

Achievable ☐ Achieving the objective should be within the team's control and influence.

Realistic ☐ Recognize factors that cannot be controlled. (Hint: Avoid making "happy path" assumptions.)

Time-bound ☐ The time period for achievement must be within the PI or sooner, so all objectives must be scoped appropriately.

Note: SMART PI Objectives are similar to Key Results in the OKR format in that they are tangible and measurable. However, the OKR format has proven less effective when applied to PI Objectives. See [OKRs](#) for more detail.

Communicating Business Value with PI Objectives

As objectives are finalized during PI planning, Business Owners collaboratively assign "business value" to each team's objectives face-to-face. The value of this conversation with the team cannot be overstated, as it communicates the strategy and context behind these weighting decisions. Business Owners use a scale from 1 (lowest) to 10 (highest) to rate each objective. They need not be "normalized" across teams; every team has some highest priority (rated 10) items.

Business value is assigned, not calculated, and serves as an input to execution considerations. Many of the team's objectives provide direct and immediate value

to the solution. Others, such as **Enablers** (for example, advances in infrastructure, development environments, and quality initiatives), allow the creation of future business value faster. All of these factors must be weighed in the final balance.

Finalize Team PI Objectives

When objectives have been made ÔSMARTer, uncommitted objectives have been identified, and business value has been established, the objectives in Figure 1 might evolve to look like those in Figure 3.

Figure 3. The team's final PI objectives with business value assigned

Commit to PI Objectives

A vote of confidence is held near the end of PI planning, where the teams commit to the PI objectives. (Uncommitted objectives are *not* included in this commitment.) However, it must be a reasonable ask for the people who do the work. Therefore, the SAFe commitment has two parts:

Teams agree to do everything reasonably in their power to meet the committed objectives

During the PI, if it's discovered that some objectives are not achievable, then the

teams agree to immediately escalate so that stakeholders are informed and corrective action can be taken

In this way, all stakeholders know that either the ART results will be achieved as planned or they will be provided sufficient notice to mitigate and take corrective action, minimizing business disruption. That's about as good as it gets because this is, after all, research and development.

Creating ART and Solution Train PI Objectives

The output of the PI planning process will be a collection of approved team PI objectives. Teams vote on the confidence level for the objectives as a set, and if confidence is high enough, the aggregate set of objectives becomes the committed ART plan. The [Release Train Engineer](#) summarizes the team objectives into the ART PI objectives in a format suitable for management communication.

The summarized objectives should be SMART, like the team PI objectives, and have uncommitted objectives. Also, like the team PI objectives, the ART PI objectives might describe business features the ART is working on, enablers, or other business or technical goals.

If the ART is part of a [Solution Train](#), the objectives are further rolled up by the [Solution Train Engineer](#), and the Solution Train PI objectives are synthesized and summarized. This is the top level of PI objectives in SAFe, and they communicate to stakeholders what the Solution Train will deliver in the upcoming PI. Figure 4 below illustrates this summary from team to ART and from ART to Solution Train PI objectives.

Business value must only be assigned to team PI objectives. The predictability metric itself is rolled up to determine predictability at a higher level.

Figure 4. Roll-up of the team, ART, and Solution Train PI objectives

Figure 4. Roll-up of the team, ART, and Solution Train PI objectives

Reduce WIP with Realistic PI Objectives

During the team PI objectives review, not everything the various business stakeholders envisioned will likely be achieved in the PI timebox. Therefore, some of the planned work will need to be reevaluated with Business Owners to gain agreement with the PI objectives.

Those lower-priority work items get moved back into the [ART Backlog](#). Decreasing excess WIP reduces overhead and thrashing and increases productivity and velocity. The net result is a feasible set of PI objectives agreed to by all business stakeholders and team members, increased efficiency, and a higher probability of delivery success.

Planning at the large solution level can be very similar; the planning of the ARTs will impact each other, pushing some work back into the [Solution Train Backlog](#) for re-evaluation in a later PI.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Reinertsen, Donald. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

Last update: 13 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
 Weekends: CLOSED

[Privacy Policy](#)

English

Scaled Agile Framework



*It's not enough
that management
commit themselves
to quality and
productivity, they
must know what it
is they must do.
Such a
responsibility
cannot be
delegated.*

ÑW. Edwards
Deming

Adjust Cookie Settings

Lean-Agile Leadership

Introduction

The Lean-Agile Leadership competency describes how Lean-Agile Leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential. They do this through leading by example, learning and modeling SAFe's Lean-Agile mindset, values, principles, and practices, and leading the change to a new way of working.

Lean-Agile Leadership is one of the seven core competencies of Business Agility, each of which is essential to achieving Business Agility. Each core competency is supported by a specific assessment, enabling the enterprise to assess its proficiency. The Measure and Grow article provides these core competency assessments and recommended improvement opportunities.

Why Lean-Agile Leaders?

An organization's managers, executives, and other leaders are responsible for the adoption, success, and ongoing improvement of Lean-Agile development and the competencies that lead to business agility. Only they have the authority to change and continuously improve the systems that govern how work is performed. Moreover, only these leaders can create an environment that encourages high-performing Agile teams to flourish and produce value. Leaders, therefore, must internalize and model leaner ways of thinking and operating so that team members will learn from their example, coaching, and encouragement.

Achieving agility throughout the enterprise is neither simple nor easy. As described below, business agility requires a new approach to leadership. It starts with leaders exemplifying behaviors that will inspire and motivate the organization to pursue a better way of working. They set the example by coaching, empowering, and engaging individuals and teams to reach their highest potential through Lean and Agile principles and practices.

In short, knowledge alone won't be enough. Lean-Agile leaders must do more than simply support the transformation: they must actively *lead* the change, participating in and guiding the activities necessary to understand and continuously optimize the flow of value through the enterprise. Lean-Agile leaders:

- Organize and reorganize around value
- Identify and reduce long queues and excess Work in Process (WIP)
- Continually focus on eliminating bottlenecks and delays

- Eliminate demotivating policies and procedures
- Inspire and motivate others
- Create a culture of relentless improvement
- Provide the space for teams to innovate

By helping leaders develop along three distinct dimensions, as illustrated in Figure 1, organizations can establish Lean-Agile leadership as a core competency:

Figure 1. The dimensions of lean-agile leadership

Figure 1. The dimensions of lean-agile leadership

These dimensions are:

1. **Leading by Example** □ Leaders gain *earned authority* by modeling the desired behaviors for others to follow, inspiring them to incorporate the leader's example into their personal development journey.
2. **Mindset and Principles** □ By embedding the Lean-Agile way of working in their core beliefs, decisions, responses, and actions, leaders model the expected norm throughout the organization.
3. **Leading Change** □ Leaders *lead* (rather than simply support) the transformation by

creating the environment, preparing the people, and providing the necessary resources to realize the desired outcomes of the change.

The following sections explore these dimensions of Lean-Agile leadership in greater detail.

Leading by Example



Setting an example is not the main means of influencing others, it is the only means.

ÑAlbert Einstein

Through their words and actions, leaders provide the organization with patterns of expected behaviors. The aggregation of those patterns determines the organization's culture, whether good or bad. The most important and effective technique for driving the cultural change needed to transform to the new way of working is for leaders to internalize and model the behaviors and mindsets of business agility so that others can learn and grow by their example.

Author Simon Sinek underscores the importance of leading by example in his book *Leaders Eat Last* [1] with the following:

The leaders of companies set the tone and direction for the people. Hypocrites, liars, and self-interested leaders create cultures filled with hypocrites, liars, and self-interested employees. The leaders of companies who tell the truth, in contrast, will create a culture of people who tell the truth. It is not rocket science. We follow the leader.

By modeling the right behaviors, leaders can transform organizational cultures from the pathological (negative, power-oriented) and bureaucratic (negative, rule-oriented) patterns of the past to the generative (positive, performance-oriented) culture that is required for the Lean-Agile mindset to flourish. Figure 2 provides a comparison of the attributes of Westrum's organizational culture model [2]. These same behaviors also build **earned authority** Ñ power gained through trust, respect, expertise, or actionÑwhich engenders greater engagement and commitment to organizational aims than positional authority. Such leaders inspire others to follow their direction and to incorporate the leader's example into their personal development journey.

Figure 2. Westrum's organizational cultures model (adapted)

Figure 2. Westrum's organizational cultures model (adapted)

As we learn more about the challenges of the digital age and the critical competencies leaders need to guide the organization to greater business agility, it's essential to understand that the best outcomes are achieved when leaders model behaviors that foster a **generative** culture.

What, then, are the behaviors that leaders should embrace to set the right example and build a generative culture? While the potential list of attributes could be long, the leader behaviors below form a solid foundation for this dimension of leadership.

Insatiable learning depicts how leaders engage in the ongoing, voluntary, and self-motivated pursuit of knowledge and growth and encourage and support the same in others.

Authenticity requires leaders to model desired professional and ethical behaviors. Acting with honesty, integrity, and transparency, they are true to themselves and their beliefs.

Emotional competence describes how leaders identify and manage their emotions and those of others through self-awareness, self-regulation, motivation, empathy, and social skills.

Courage is essential for leaders to guide their organizations through the rapidly changing dynamics of the digital age. It requires leaders to embrace vulnerability, take appropriate risks, and engage in difficult but necessary conversations to challenge the status quo.

Growing others encourages leaders to provide each employee's personal, professional, and technical guidance and resources to assume increasing levels of responsibility and decision-making.

Decentralized decision-making moves the authority for decisions to where the information is; it prepares teams to make decentralized decisions by investing in their technical competence and providing organizational clarity with decision guardrails. [3].

These behaviors are critical to leading in the digital age, where business growth is fueled by an increased shift to instant access to information, entertainment, social and business connections, products, and services, predominantly via mobile and smart devices. This modern context requires mindsets and skills that can be significantly different from the past patterns of working and leading that were successful. If leaders fail to adapt to the rapidly changing demands of a digital economy, their organizations will be significantly disadvantaged.

Mindset and Principles



ÒThe basic tenets of Lean challenge many of the aspects of traditional management theory and calls for a mindset that is foreign to most executives.Ó

Ñ Jacob Stoller, author of *The Lean CEO: Leading the Way to World-Class Excellence*

Stoller's quote is a reminder that traditional management practices are insufficient for the changes needed to achieve business agility. Instead, the Lean enterprise depends on what Toyota calls *Lean-thinking manager-teachers*. These leaders understand Lean thinking and principles and teach them to others in their everyday work activities. This is integral to *who they are* and *what they do*. It

informs every aspect of their approach to helping teams throughout the organization work in a Lean and Agile manner as the expected norm.

But what if leaders don't have that mindset yet? What is a "mindset", and how can a mindset be changed?

Mindset Awareness and Openness to Change

A *mindset* is simply the mental lens through which we view the world around us. It is how the human brain simplifies, categorizes, and interprets the vast amounts of information it receives daily. We form our mindsets through a lifetime of structured learning (classes, reading) and unstructured lessons (life events, work experience). They reside in the subconscious mind and manifest themselves as deeply held beliefs, attitudes, assumptions, and influences. Consequently, individuals are often unaware of how their mindsets influence how they carry out their responsibilities and interact with others. For example, many leaders develop beliefs through business school training and on-the-job experience grounded in legacy waterfall, stage-gate, and siloed ways of working.

So how can mindsets be changed? It begins with the awareness of how one's current mindset was formed. It's also vital to cultivate the belief that mindsets can be developed and improved (a "growth" mindset, as illustrated in Figure 3). Leaders must remain open to the possibility that existing mindsets based on traditional management practices must evolve to guide the organizational change required to become a Lean enterprise. [4]

Figure 3. Adopting a new mindset requires a belief that new abilities can be developed with effort

Developing a New Mindset

With an increased awareness of current mindsets and an openness to doing the work required to change them, the question becomes, “Change them to what?” Leading the organization through the transformation needed to achieve business agility requires a mindset that reflects the core values and principles of Lean, Agile, and SAFe. This is developed by gaining intimate knowledge and *applying* these values and principles. It is reflected in how leaders routinely reference Lean-Agile principles and practices as part of carrying out their responsibilities, how they coach and mentor these behaviors in others, and how they promote Lean-Agile practices as the default way of working throughout the organization.

Let’s take a closer look at the three key elements that form the foundation of this new mindset: SAFe Core Values, the Lean-Agile Mindset, and SAFe Principles.

SAFe Core Values

SAFe's essential ideals and beliefs are defined by alignment, transparency, respect for people, and relentless improvement. Leader behaviors play a critical role in communicating, exhibiting, and emphasizing these values and how they guide the organization's journey to embracing agility.

Here are some suggestions for how leaders reinforce these values:

Alignment □ Communicate the vision, mission, and strategy, and connect them to the portfolio work through well-formed strategic themes. Help organize the value stream and connect strategy to execution through the portfolio vision, lean budgets, and epics. Provide relevant briefings and participate in [PI Planning](#). Help with backlog visibility, review, and preparation; regularly check for understanding.

Transparency □ Visualize all relevant work. Take ownership and responsibility for errors and mistakes, using them as learning moments. Admit missteps while supporting others who acknowledge and learn from theirs. Never punish the messenger. Instead, celebrate learning. Create an environment where the facts are always friendly and transparent. Ensure everyone throughout the organization has ready access to needed information.

Respect for people □ Treat people with authentic trust and respect. Value diverse opinions and viewpoints. Have genuine care and concern for the growth and development of others by providing coaching, mentoring, training, and enriching experiences. Extend respect to internal and external customers as well as partners and suppliers.

Relentless improvement □ Provide priority, visibility, and resources to improvement efforts. Promote a problem-solving culture of "everyone improving all the time." Encourage consistency in conducting retrospectives and following through with improvements identified as part of the Problem-Solving Workshop during Inspect & Adapt (I&A). Protect time and space for innovation, especially during the Innovation and Planning (IP) iteration.

Lean-Agile Mindset

The genesis of SAFe was to develop guidance for enterprises on how to apply the principles and practices of Lean and Agile in the world's largest organizations. A [Lean-Agile Mindset](#) requires leaders to learn, embrace, and model both Lean and Agile in their behaviors and support adoption by the enterprise. Figure 4 illustrates the key concepts of each discipline.

Figure 4. Lean Thinking principles and the values of the Agile Manifesto

Figure 4. Lean Thinking principles and the values of the Agile Manifesto

Lean Thinking Ⓜ Lean is a set of principles and practices for efficient manufacturing and operations that grew out of the Toyota Production System developed in post-WWII Japan. It focuses on problem-solving and continuous improvement to increase quality and eliminate waste. Adapted to product development by Leffingwell [5], Poppendieck [6], and others, the principles of Lean Thinking illustrate the goal of delivering value by *precisely specifying value by product, identifying the Value Stream for each product, making value flow without interruptions, letting the customer pull value from the producer, and pursuing perfection*. Leaders create the environment for Lean Thinking by internalizing these principles and exemplifying them in their words, actions, and decision-making process.

Agile Ⓜ Agile was born from a collaboration of 17 thought leaders in software development who met in 2001 to seek alternatives to the documentation-driven, heavyweight software development processes that were common at the time. It includes four values (shown in Figure 4) and twelve principles as reflected in the Agile Manifesto. Agile is known for delivering iterative and incremental value in the

form of working software by promoting face-to-face interaction frequently between developers, customers, and cross-functional, self-organizing teams. Agile has since been adapted and embraced in many non-software development contexts.

The [Lean-Agile Mindset](#) article describes how Lean and Agile are at the heart of SAFe and are supported by many articles in the Framework that explain how to implement Lean-Agile practices at scale. Many great courses, books, websites, and videos form a rich set of resources that Lean-Agile leaders should explore to deepen their understanding.

SAFe Principles

SAFe is based on ten immutable, underlying principles. These tenets and economic concepts inspire and inform the roles and practices of SAFe, influencing leader behaviors and decision-making.

The principles are:

Figure 5. The SAFe Lean-Agile Principles

Each is necessary to experience the personal, business, and economic benefits of applying SAFe. Moreover, these principles work together as a system; each informs the others, and the whole is far greater than their sum individually. Lean-Agile

leaders embrace these principles and routinely demonstrate and apply them to carry out their organizational responsibilities. Review the SAFe Principles articles for a more in-depth discussion of each principle.

Leading Change

Being a Lean-Agile leader provides the thought processes and practical tools needed to guide the enterprise to achieve business agility. The benefits of delivering value in the shortest sustainable lead time, creating flow, and producing customer delight—all with happy, engaged employees—are clear. It's also clear that the new way of working for many organizations represents a quantum shift in culture and practice from the traditional paradigms of the past. In other words, the transformation required to adopt SAFe inevitably leads to significant organizational change.

Here again, the role of the Lean-Agile leader is critical. Successful organizational change requires leaders who will *lead* the transformation (rather than simply “support” it) by creating the environment, preparing the people, and providing the necessary resources to realize desired outcomes. Research shows clear correlations between the leader behaviors described in the “Leading by Example” section of this article and the success of organizational change driven by Agile and DevOps initiatives. Other researchers found that these leader behaviors influence employees’ commitment to supporting the change more than simply following a prescriptive change model [7, 8].

Lean-Agile leaders drive the change process by developing and applying the following skills and techniques:

Change vision occurs when leaders effectively communicate *why* change is needed and do so in ways that inspire, motivate, and engage people to buy into the change with a sense of urgency.

A powerful coalition for change is formed when a “volunteer army” of individuals from multiple levels, across silos, and with diverse perspectives are empowered to contribute to and help overcome barriers to implementing the change.

Change leadership is the ability to positively influence and motivate others to engage in organizational change through the leader’s personal advocacy and drive. It includes producing and celebrating short-term wins, reinforcing the change until the desired outcomes are achieved, and anchoring the change in the organization as the “new normal.”

Psychological safety occurs when leaders create an environment for risk-taking that supports change without fear of negative consequences to self-image, status, or career.

Training the new way of working ensures that everyone is trained in the values, principles, and practices of Lean and Agile, including a commitment by leaders to their training so they can lead by example.

Sound organizational change management (OCM) practices are still important and highly recommended in a SAFe transformation. One of the most respected voices in OCM, Dr. John Kotter, in his most recent research, has described the Ôeight acceleratorsÕ for implementing successful change as [9, 10]:

1. Create a sense of urgency
2. Pull together the guiding team
3. Develop the change vision and strategy
4. Communicate for understanding and buy-in
5. Empower others to act
6. Produce short-term wins
7. DonÕt let up
8. Institute change

Dr. Kotter goes further in describing four change leadership principles that can help unlock the full potential of the eight accelerators:

Management + Leadership. To capitalize on windows of opportunity, leadership must be paramount Ð and not just from one executive. ItÕs about vision, action, innovation, and celebration, as well as essential managerial processes.

ÔHave toÕ + ÔWant toÕ. Those who feel included in a meaningful opportunity will help create change in addition to their normal responsibilities. Existing team members can provide the energy Ð if you invite them.

Head + Heart. Most people arenÕt inspired by logic alone, but rather by the fundamental desire to contribute to a larger cause. Extraordinary results are possible if you can give greater meaning and purpose to your effort.

Select Few + Diverse Many. More people need to be able to make change happen Ð not just carry out someone elseÕs directives. Done right, this uncovers leaders at all levels of an organization, ones you never knew you had.

These values and practices require the active participation of the leaders driving the change. But even this is not enough. Heath and Heath note in their book on change [11] that leaders Ôneed to script the critical movesÕ essential to accomplish the change.

The SAFe Implementation Roadmap

Based on experience and these insights from the field of organizational change management, the SAFe Implementation Roadmap article series guides leaders on this particular journey, as summarized in the [Implementation Roadmap](#) article and Figure 6 below.

Figure 6. The SAFe Implementation Roadmap

The SAFe implementation roadmap is described in a series of 12 articles that align with Kotter's blueprint for change. For example, the sense of urgency is often established in the many conversations that lead to an organization "reaching the tipping point" and deciding to "go SAFe." The next recommended action is to train a core group of Lean-Agile change agents and leaders who will form the powerful guiding coalition. The pattern continues throughout the roadmap, designed to incorporate successful organizational change lessons into the SAFe transformation model. This roadmap helps leaders "know the way" as they drive for successful change. The roadmap also highlights Scaled Agile's *Leading in the Digital Age* series designed to better prepare leaders to lead the implementation.

Role of the SAFe Practice Consultants

Even with Lean-Agile leaders and sound organizational change strategies, observations from many SAFe implementations indicate that a significant cadre of change agents and experienced coaches is also needed. While every leader plays a

part in producing the change, SAFe Practice Consultants (SPCs) are trained and equipped specially for this task. SPCs' training, tools, courseware, and intrinsic motivation play a critical role in successfully implementing and sustaining a SAFe transformation.

Summary

Implementing SAFe is not just any change; it's a shift to persistently and relentlessly improving business agility, all based on the fundamentals of Agile and Lean. It requires managers, executives, and other leaders who understand how to lead, sustain and accelerate the transformation to a new way of working.

Leaders alone have the authority to change and continuously improve the systems that govern how work is performed. Only they can create an environment that encourages high-performing Agile teams to flourish and produce value. Leaders, therefore, must internalize and model leaner ways of thinking and operating so the rest of the organization will learn from their example, coaching, and encouragement.

Effective leadership ultimately provides the foundation for the adoption and success of Lean-Agile as the new way of working, and mastery of the competencies that lead to business agility.

Learn More

[1] Sinek, Simon. *Leaders Eat Last*. Penguin Random House LLC, 2017.

[2] Westrum, Ron. *A topology of organizational cultures*. 2004. Quality and Safety in Health Care, 13(Suppl II):ii22–ii27. doi: 10.1136/qshc.2003.009522

[3] Marquet, David. *Turn the Ship Around*. Penguin Group, 2013.

[4] Dweck, Carol S. *Mindset: The New Psychology of Success*. Random House Publishing, 2007.

[5] Leffingwell, Dean. *Agile Software Requirements*. Addison-Wesley, 2011.

[6] Poppendieck, Mary, and Tom Poppendieck. *Implementing Lean Software Development: Concept to Cash*. Addison-Wesley, 2006.

[7] Mayner, Stephen. *Transformational leadership and organizational change during agile and devops initiatives*. ProQuest, 2017.

[8] Herold, David M., Donald B. Fedor, Steven Caldwell, and Yi Liu. *The effects of transformational and change leadership on employees' commitment to change: a multi-level study*. Journal of Applied Psychology, Volume 93, 2008.

[9] Kotter, John P. *Accelerate: Building Strategic Agility for a Faster-Moving World*. Harvard Business Review Press, 2014.

[10] Kotter, John P. *Change: How Organizations Achieve Hard-to-Imagine Results in Uncertain and Volatile Times*. Harvard Business Review Press, 2021.

[11] Heath, Chip, and Dan Heath. *Switch: How to Change Things When Change Is Hard*. The Crown Publishing Group, 2010.

Last update: 2 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)
[Blog](#)

TRAINING

[Course Calendar](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL
[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

[About Certification](#)

Weekdays: 9am to 5pm

[Become a Trainer](#)

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



In a time of unprecedented technological advances and global change, Federal leaders, managers, and front-line employees must not only be aware of the changes to work, workforce, and workplace, but must also be able to identify weak signals, anticipate trends, and plan for

Adjust Cookie Settings

*the inevitable
changes.*

~2022 Federal
Workforce Priorities
Report [1]

SAFe for Government

SAFe for Government is a set of success patterns that help public sector organizations achieve better solution development outcomes by implementing SAFe Lean-Agile values, mindset, principles, and practices.

The foundations of Lean and Agile thinking have led to higher success rates versus waterfall methods in software and systems development in the private sector. Government programs are starting to experience similar results using these same patterns. However, government agencies must address unique challenges in Lean-Agile transformations. The recommendations and best practices in SAFe for Government provide specific guidance to address these concerns.

The government-specific SAFe guidance is provided in the following series of nine articles:

- Building on a solid foundation of Lean-Agile values, principles, and practices
- Creating high-performing teams of teams of government and contractor personnel
- Aligning technology investments with agency strategy
- Transitioning from projects to a lean flow of epics
- Adopting Lean budgeting aligned to development value streams
- Applying Lean estimating and forecasting in cadence
- Modifying acquisition practices to enable Lean-Agile development and operations
- Building in quality and compliance
- Adapting governance practices to support agility and lean flow of value

The remainder of this summary article sets the context for why the principles and practices of Lean-Agile and SAFe are especially relevant for government agencies and provides a summary of each of the individual articles in the series.

(For more information on this topic, including links to pages containing a wealth of additional resources and videos, check out our [Agile in US Government](#) page on [ScaledAgile.com](#).)

Details

Why SAFe for Government?

Lean-Agile and DevSecOps practices are gaining interest among the leaders responsible for the largest, most complex systems built for government worldwide. Much of that interest is driven by the internal and external forces changing how government agencies provide services to citizens, the government workforce, and warfighters. Governments are concerned with mission agility. While managing limited resources and funding is a concern, governments are not profit-driven but instead measure success as the value of impact that a solution provides. In this regard, governments resemble very large, complex non-profit organizations striving to run the business of operating as effectively and efficiently as possible. These imperatives drive:

- The need for business (mission) agility
- Impact of digital transformation
- The rise of social media and instant access to information about IT spending
- Increasing citizen expectations
- Technical debt and antiquated systems driving IT modernization initiatives
- Rapid changes in defense systems and the global cyber threat environment
- Fostering an Agile organization and growth mindset for the government workforce

Like for-profit organizations, the government is increasingly dependent upon technology. Yet, the traditional approaches to developing and sustaining solutions over the last 60 years have proven insufficient when developing modern technology-based capabilities. Agile practices have shown promise. However, the size and complexity of government systems, ranging from an unemployment benefits website for French citizens to large cyber-physical Sentinel missile systems, require more than team-level Agile and DevSecOps practices can provide.

Background of Agile Adoption in the U.S.

Federal Government

Interest in Lean-Agile methodologies for government technology development programs has increased exponentially in the U.S. since 2022. In July of that year, the U.S. General Accounting Office (GAO) issued a report recommending specific practices for Agile development, along with 14 unique challenges to Agile adoption in government. [2] That same year, the U.S. Office of Management and Budget (OMB) directed agencies to change their procurement practices from bloated long-

term projects to a more modular contracting approach aligned with an iterative development model. [3] Although these were positive signs that the decades-old commitment to waterfall processes was relenting, agencies were still slow to adopt a different way of working. Figure 1 shows a timeline of the significant events that have driven Lean-Agile adoption in the U.S. federal government.

Figure 1. Significant events that have driven Lean-Agile adoption in the U.S. government since 2010

The troubled launch of the U.S. Healthcare.gov website in the latter part of 2013 increased the interest in adopting Agile almost overnight. This site allows citizens to obtain health insurance as part of the Affordable Care Act (ACA). For weeks, the website's initial difficulties were highlighted in the national news, exposing many weaknesses in the traditional development practices common to government technology programs. [4] These events' public focus on government IT programs drove increased openness to adopt more modern development practices. In an analysis of Agile adoption in the U.S. Federal Government published by Deloitte in

2017, [5] the percentage of federal IT projects that report using Agile or iterative processes has grown dramatically since 2012, as shown in Figure 2.

Figure 2. Percentage of federal programs that report using an Agile or iterative development model

The increased interest in Lean, Agile, and DevOps accelerated when two new U.S. government agencies, 18F and U.S. Digital Service, were created to attract talent from industry to help bring modern, Silicon Valley-like practices to federal IT programs. The [Digital Services Playbook](#) and the [TechFAR Handbook](#) were two early resources they provided to help leaders in government programs understand how to modernize development practices and adjust the acquisition process to support Agile contracts. The number of additional government-authored resources on Agile adoption has grown significantly, as has the number of published success stories of federal programs getting better results after their transformation to Lean-Agile practices. One agency, the Department of Homeland Security, has made Agile the formal standard for software development. In September 2020, the Government

Accounting Office (GAO) released its draft of [The GAO Agile Assessment Guide: Best Practices for Agile Adoption and Implementation](#) for government programs using Agile practices. Congress continues to direct increased training in Agile as well as authorizing the modernization of acquisition practices to support Agile through legislation such as the annual National Defense Authorization Act (NDAA).

Agile Adoption in the Global Governments

Similar trends are being experienced in systems development for state and local governments and governments in countries across the globe. The UK government has been engaged for several years in an Agile transformation effort that spans hundreds of Agile teams across its many departments and agencies. PTMle Emploi (French employment agency), the Dutch Tax Authority, and the Australian Post (postal service) are examples of global government agencies using SAFe to guide their transformations to Lean-Agile at scale. In the case of PTMle Emploi, their transformation has resulted in better on-time delivery of employment benefits and increased satisfaction with the agency's services by both hiring businesses and job seekers.

Government as a Lean Enterprise

Increasingly, government agencies are challenged by the same forces of change driving their commercial counterparts to accelerate Lean-Agile transformations. The need for business (mission) agility, digital disruption, globalization, ever-increasing cyber threats, aging legacy systems, and increasing dependency on technology for business and mission success are just a few of the factors that are equally concerning for both government and industry.

The hallmark of a Lean Enterprise is the ability to deliver the best quality and value in the sustainably shortest lead time. SAFe provides guidance by describing the success patterns that help organizations achieve these competencies. The evidence provided by the many case studies of SAFe implementations in government agencies suggests that these competencies apply to public sector organizations as they do to commercial enterprises.

How SAFe Enables Lean-Agile and DevOps in Government

“In 10 months’ time, we have turned a failing effort into a success story for the warfighter, for our organizational culture, and for the taxpayer. We could not have executed a turnaround this fast without SAFe.”

Ñ Scott Keenan, JLVC Program Manager, Joint Staff (U.S. Department of Defense)

SAFe Adoption in Government is Growing

Many government technology programs are large and complex, involving hundreds (sometimes thousands) of practitioners. Solutions are built by multiple teams of teams that need to:

- Plan and work together
- Manage cross-team dependencies
- Integrate frequently
- Demonstrate working software and systems iteratively
- Share learning for relentless improvement

Often, these large solutions include small groups of government employees working closely with large numbers of contractor personnel. Multiple suppliers on different contracts may work on the same program across dispersed geographies. High assurance and compliance requirements, burdensome governance regulations, and abnormally long acquisition lead times further complicate government projects.

Most of these challenges are also common in large commercial organizations. Companies in the Global 1000 have used Lean-Agile to build core banking systems, satellites, farming combines, clinical and financial systems for health networks, and much more. For these organizations, SAFe has emerged as the leading Framework of practices for Lean-Agile at scale; the positive results have been documented in published case studies. Many government agencies have adopted SAFe as the process model for technology development for the same reasons as their commercial counterparts. Practitioners who have used SAFe in both contexts have reported that *there are far more similarities in development between industry and government than differences*. As shown in Figure 3, SAFe is now being used in hundreds of programs across a large number of government agencies.

Figure 3. Global government agencies with programs using SAFe as their development Framework

The Government has Unique Challenges Along the Path to Lean-Agile Adoption

Despite the increased momentum toward Lean-Agile and using SAFe, several barriers delay widespread adoption. The most frequently cited challenges include:

1. Poor implementations of Agile in the past created a reluctance to try again
2. Waterfall-centric governance and lifecycle policies that are not easily changed
3. An acquisition workforce that lacks experience with Agile contracts and Lean contracting practices
4. Project orientation, versus a continuous-flow-of-value mindset, is deeply ingrained in government culture
5. Long acquisition lifecycles create tremendous waste and delays in delivering value
6. The lack of a common enterprise Lean-Agile Framework leads to a limited synergy between programs

Although these problems look similar to those faced by commercial companies, the organizational context, culture, and governmental authorities in the public-sector environment are unique. Government acquisition processes and laws are intended to create a fair playing field among potential providers, but they can also create bureaucracy and delay unlike anything the private sector experiences. In addition, government agencies do not have the competitive market dynamic and profit motive that drives rapid change and innovation in a commercial environment. Instead, legislative bodies typically provide funding in politically-charged annual appropriations processes that move slowly. Even a government technology program's concept of "value" is often difficult to conceptualize and measure.

The Solution Ð Government-Specific Guidance for SAFe

Because factors in government technology development are specific to these environments, specialized guidance is required to help change agents as they lead transformations. The following links connect to articles addressing the most common challenges of adopting SAFe in government programs and the best practices that support transitioning to a Lean-Agile model.

Building on a solid foundation of Lean-Agile values, principles, and practices.

Waterfall ways of thinking and processes are deeply ingrained in government technology programs. Simply mimicking practices like the Team Sync and working from backlogs will not achieve agency agility. Government leaders and practitioners and their industry partners must understand how and why Lean-Agile fundamentally differs from past approaches to technology development.

Creating high-performing teams of teams of government and contractor personnel.

Lean-Agile development is a team sport. Teams on government programs often include a combination of government and contractor personnel. Too often, the relationship between these entities is more antagonistic than cooperative. This friction inhibits the development of high-performing teams and, ultimately, the rapid delivery of valuable quality solutions.

Aligning technology investments with agency strategy.

Government technology programs can be proposed, approved, and funded for various reasons. Technology investments are often made on the basis of assumed funding of past initiatives without the benefit of a periodic review of holistic portfolios of systems. The best use of limited development funds is achieved by ensuring that priorities are aligned and in harmony with the strategic imperatives of the agency.

Transitioning from projects to a lean flow of epics.

The start-stop nature of the Ôproject metaphorÕ for technology development leads to committing to point solutions too early in the process when there are too many unknowns and committing to delivering all planned features even if completing the highest priority features provides most of the value. Projects also encourage moving people to the work instead of flowing work to teams through single-piece flow backlogs. The Lean alternative is to organize large efforts into Epics and manage them in flow from a prioritized backlog built by long-lived teams of teams.

Adopting Lean budgeting aligned to development value streams.

Along with the shift from projects to a lean flow of Epics, there is a change in how to budget. Instead of funding individual pieces of work, budgets fund Ôthe factoryÕ that can build whatever the agency needs based on priorities. Since priorities can continuously shift and change, this approach avoids the wasted expense and delay of change orders or entirely new solicitations to shift investments from one

initiative to another, enabling agency agility.

Applying Lean estimating and forecasting in cadence. Lean-Agile practices recognize that traditional estimating and forecasting techniques often fail because there are too many unknowns when building entirely new systems. These traditional techniques also lock programs into the one right answer created up front and used to define contractual terms with multiple vendors. Lean estimating and forecasting techniques are lightweight and provide needed adaptability to constantly changing conditions while still providing critical reporting and accountability.

Modifying acquisition practices to enable Lean-Agile development and operations. Fewer barriers to Lean-Agile adoption in government are as daunting as the legacy acquisition process. Contracting officers depend on tried and true boilerplate language for each new acquisition firmly grounded in waterfall terms and conditions. Programs cannot be genuinely Agile if contractors are required to perform in ways that run counter to Lean-Agile values and principles. Transforming to a new way of development requires new templates for contracting officers that allow vendor partners also to be Lean-Agile.

Building in quality and compliance. The goal of Lean-Agile is to create a continuous and sustainable flow of value in technology solutions. Continuous flow is defeated if our governance processes, such as verification and validation, remain big batch activities performed only at the end of a project. Changing the way the development team works is only part of the answer. Every aspect of the technology value stream, including operations and governance, must also work in flow in smaller batches. This builds quality and compliance into the solution versus inspecting it as part of one big batch before release.

Adapting governance practices to support agility and lean flow of value.

Traditional governance standards for technology development are also deeply rooted in a waterfall model. They require programs to plan everything up front, commit to and budget for one right solution before the work begins, provide detailed project plans for unknown work, pass through arbitrary stage gates, and much more. Lean-Agile methods serve the original intent—*to provide sufficient oversight to ensure the delivery of mission-enabling capabilities within a reasonable time and cost—but through the use of alternative processes and artifacts that support a continuous flow of value.*

NOTE: These recommendations for Lean-Agile adoption in government do not require a different version of SAFe or suggest modifying SAFe terms and practices to fit government protocols. Experienced *practitioners in government services have reported that they achieve the best results when the SAFe model and*

terminology are used without modification. Using the terminology native to SAFe allows program personnel to take advantage of the classes, articles, books, forums, and other information sources necessary to implement SAFe practices successfully. The articles behind each of the practices above explain the specific patterns used by government programs to overcome the most common concerns when transitioning a federal IT initiative to SAFe.

Summary

Government adoption of Lean-Agile has accelerated to the point where most programs (80%+ in the U.S. according to a Deloitte study) are using some form of Agile or iterative development. However, Agile practices are often limited to development teams and do not address the program and portfolio challenges of strategy alignment, budgeting, project-centric planning, acquisitions, governance, compliance, and more. Agencies also lack a common language and set of enterprise-wide practices to create synergies between programs and practitioners. With this SAFe for Government guidance, supported by the corresponding SAFe for Government course, agency leaders will have the tools to overcome common barriers to SAFe adoption and Lean-Agile practices, enabling better results.

Learn More

[1] 2022 Federal Workforce Priorities Report. <https://www.opm.gov/policy-data-oversight/human-capital-management/federal-workforce-priorities-report/2022-federal-workforce-priorities-report.pdf>

[2] *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods.* General Accounting Office (GAO), July 2012.
<https://www.gao.gov/products/GAO-12-681>

[3] *Contracting Guidance to Support Modular IT Development.* Office of Management and Budget (OMB), June 2012.
<https://obamawhitehouse.archives.gov/blog/2012/06/14/greater-accountability-and-faster-delivery-through-modular-contracting>

[4] Brill, Steven. *Code Red: Inside the nightmare launch of Healthcare.gov and the team that figured out how to fix it.* Time, March 10, 2014.
<http://content.time.com/time/subscriber/article/0,33009,2166770-2,00.html>

[5] Viechnicki, Peter, and Mahesh Keikar. *Agile by the Numbers: A data analysis of*

Agile development in the US federal government. May 5, 2017.

<https://www2.deloitte.com/insights/us/en/industry/public-sector/agile-in-government-by-the-numbers.html>

[6] Pawlikowski, Ellen. *USAFO's Pawlikowski: DoD Use of Agile Software Development Critical*. <https://youtu.be/nQUplJVjql>

[7] *The Digital Services Playbook*. US Digital Service. <https://playbook.cio.gov/>

[8] *The TechFAR*. US Digital Service. <https://techfarhub.cio.gov/handbook/>

[9] *The TechFAR Hub*. US Digital Service. <https://techfarhub.cio.gov/>

[10] *Modular Contracting*. 18F. <https://modularcontracting.18f.gov/>

[11] *Digital IT Acquisition Professional Training*. Federal Acquisition Institute. <https://techfarhub.usds.gov/get-started/ditap/>

[12] *Agile Acquisitions 101*. Federal Acquisition Institute. <https://www.fai.gov/media-library/item/agile-acquisitions-101>

[13] Eggers, William D. *Delivering on Digital: The Innovators and Technologies that are Transforming Government*. Deloitte University Press, New York, 2016.

[14] *The GAO Agile Assessment Guide: Best Practices for Agile Adoption and Implementation*. <https://www.gao.gov/products/GAO-20-590G>

[15] *Science & Tech Spotlight: Agile Software Development*. General Accounting Office (GAO), Sept 2020. <https://www.gao.gov/products/gao-20-713sp>

[16] William D. Eggers, Allan Mills, Hans Verheggen and Carsten Joergensen. *Government Trends 2021- Global Transformation Trends in the public sector*. March 2021. <https://www2.deloitte.com/us/en/insights/industry/public-sector/government-trends/2021/agile-at-scale-in-government.html>

[17] DoD Software Modernization Strategy. U.S. Department of Defense. Feb 2022. <https://www.defense.gov/News Releases/Release/Article/2924187/dod-software-modernization-strategy-approved/>

Last update: 15 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The use of KPIs is meant to improve and transform the organizational performance.

—Pearl Zhu,
Performance Master: Take a Holistic Approach to Unlock Digital Performance

Value Stream KPIs

Adjust Cookie Settings

Value Stream Key Performance Indicators (KPIs) are the quantifiable measures used to evaluate how a value stream performs against its business objectives.

Development Value Stream KPIs help close the feedback loop from:

[Strategic Themes](#) through the Lean Budgeting process

Epics (portfolio initiatives) established in the [Portfolio Kanban](#)

Business outcomes created by portfolio investments in [Development Value Streams](#)

Details

[Principle #9 - Decentralize decision-making](#) is a critical mindset for operating a SAFe portfolio. Applying this principle rewards the business and its employees greatly. However, it challenges many aspects of traditional governance and operations. For example, the SAFe [Lean Budgeting](#) approach simplifies financial management, empowers decentralized decision-making, and increases the flow of value from the [Enterprise](#) to its customers.

But how does the enterprise know if those decisions were optimal? One solution is creating clear targets for expected outcomes rather than identifying and managing the projects (or tasks) needed to achieve the results. Each development value stream defines a set of quantifiable measurements or *Key Performance Indicators* (KPIs) that evaluate their ongoing investment.

Strategic Themes Inform Value Stream KPIs

Each portfolio has a set of Strategic Themes that connect the enterprise strategy to the [Portfolio Vision](#). They reflect the intended business outcomes of strategy, investment, and portfolio vision. As the strategic themes article describes, formulating themes as [Objective and Key Results \(OKR\)](#) is the preferred approach for defining the primary high-level business outcomes. The entire set of results reflects the overall business intent of that portfolio.

Therefore, OKRs are delivered or supported by the portfolio's development value streams, individually or together. Consequently, OKRs are likely related directly or indirectly to KPIs, as Figure 1 illustrates.

Figure 1. Value stream KPIs are derived from strategic themes and local concerns

Figure 1. Value stream KPIs are derived from strategic themes and local concerns

Figure 2 shows an example of a strategic theme, *“Increase annual recurring revenue”*, expressed as an OKR for the Autonomous Solutions portfolio. It has three key results to measure this objective:

- Increase account renewals from 75% to 90%
- Increase lead conversion from 20 to 40%
- Optimize customer acquisition process

Some key results are measured by one value stream KPI, like the example in Figure 2. For instance, the key result, “Increase account renewals from 75% to 90%,” is measured by the value stream KPI for the number of account renewals. Sometimes, key results require multiple KPIs to measure them. However, other relationships are less clear. They may cross values streams, summarize results of value stream coordination, or influence outcomes in some other way that does not directly drive a specific KPI.

Figure 2. Value stream KPIs can help measure the key results for an OKR

Figure 2. Value stream KPIs can help measure the key results for an OKR

Local Context Informs Development Value Stream KPIs

In addition, as Figure 1 illustrates, while OKRs are indeed significant concerns of the portfolio, each development value stream has a local context. This local context means some value stream KPIs are derived from a specific value stream's business objectives. As a result, they may not trace directly to a strategic theme's *key result*.

SAFe's development value streams enable the *Operational Value Streams* (OVS) to achieve targeted business outcomes. Therefore, the operational value streams may inform the KPIs for the local development value stream. These KPIs may differ from one value stream to another within the same portfolio. For example:

Some development value streams support revenue generation directly, making

revenue a likely KPI. Other metrics—such as operating profit margin, market share, or solution usage—may provide additional insights.

Other development value streams are developing emerging offerings. Although the return on investment (ROI) would seem to be an obvious KPI, ROI is a lagging economic indicator and may not help measure early-stage investment. Instead, [Innovation Accounting](#), non-financial KPIs, offer faster feedback.

Some development value streams are primarily cost centers, serving internal operational value streams. In this case, other measures may be more relevant, including:

- [Customer](#) and [Business Owner](#) satisfaction

- Absolute costs and ratios for new development versus maintenance

- Net promoter score

- Output measures like [Feature](#) cycle time

Tips for Establishing KPIs

Since KPIs are specific to the type of development value stream, SAFe can only offer general guidance. However, the following tips and examples provide universal recommendations for establishing KPIs:

- Good KPIs focus on objective, quantifiable, and measurable business outcomes

- Some development value stream KPIs should focus on process measures such as speed, quality, flow, and self-assessments (see [Measure and Grow](#) for more).

- Trending data allows you to predict better what will happen based on history.

- Ratio data enables comparison. A sudden spike or a long-term trend becomes visible if you compare a daily metric to the same one over a month.

- Limit the number of KPIs tracked for each value stream. Typically, four to seven provides a good set of metrics.

Often, the most meaningful metrics to assess the *development* value stream’s business outcomes are associated with their *operational* value streams. For example, Figure 3 highlights some KPIs appropriate for various operational value streams. These metrics tell the story of how well the development value stream produces business results.

Figure 3. Example KPIs for different types of value streams

Figure 3. Example KPIs for different types of value streams

Closing the Loop on Lean Budgeting

Development value stream KPIs play a critical role in aligning strategy and execution. While [Agile Teams](#) are empowered and responsible for implementing the work, investment strategy rests with those responsible for ensuring sufficient capital is available to run the business.

Funding is also needed to pay [Suppliers](#), general and administrative costs, and all the costs associated with a portfolio of investments. That includes an obligation to provide an appropriate economic return to stakeholders. Strategy formulation and allocation of the enterprise's capital is a critical financial responsibility. Moreover, it's an ethical and moral obligation that cannot be left to chance or delegated beyond the appropriate level of responsibility and accountability.

Therefore, portfolio investments must be planned, allocated, and tracked, regardless of the business type or size. That is the role of [Lean Portfolio](#)

Management (LPM), which benefits significantly from value stream KPIs.

Learn More

[1] Doerr, John. *Measure What Matters*. Penguin Publishing Group, 2018.

Last updated: 1 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

English

Scaled Agile Framework



Simplicity—the art of maximizing the amount of work not done—is essential.

—The Agile Manifesto

Essential SAFe

Essential SAFe provides the minimal elements necessary for Agile Release Trains to deliver solutions and is the simplest starting point for implementation.

As a common SAFe starting point, Essential SAFe excludes Enterprise Solution Delivery and Portfolio concerns.

The [Agile Release Train \(ART\)](#) is the heart of Essential SAFe. ARTs are virtual organizations formed to span functional boundaries, eliminate unnecessary handoffs and steps, and accelerate value delivery by adopting SAFe's [Lean-Agile Principles](#) and practices.

The Essential SAFe configuration (Figure 1) includes the following constructs:

The Foundation element, which includes the [Lean-Agile Mindset](#), [Core Values](#), [SAFe Principles](#), [Implementation Roadmap](#), the role of the [SPC](#), and two core competencies:

[Lean-Agile Leadership](#) ☐ Describes how Lean-Agile Leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential.

[Continuous Learning Culture](#) ☐ Describes a set of values and practices that continually encourage individuals and the enterprise as a whole to increase knowledge, competence, performance, and innovation.

Two delivery core competencies:

[Team and Technical Agility](#) ☐ Describes the critical skills and Lean-Agile principles and practices that high-performing Agile teams and Teams of Agile teams use to create high-quality solutions for their customers.

[Agile Product Delivery](#) ☐ A customer-centric approach to defining, building, and releasing a continuous flow of valuable products and services to customers.

The essential level roles, artifacts, and events

A minimal spanning palette

Figure 1. Essential SAFe

Figure 1. Essential SAFe

Essential SAFe provides a starting point for Implementing SAFe, including the ten critical success factors needed for an ART to realize the majority of the Framework's benefits (see the last section of this article.)

Details

Essential SAFe provides the relevant roles, artifacts, events, and mindset for ARTs to deliver one or more desirable, feasible, viable, and sustainable Solutions. The ART's long-lived, flow-based, self-organizing nature powers SAFe and ultimately enables [Business Agility](#). ARTs are typically virtual, spanning organizational and geographic boundaries; others follow a line of business or product line management reporting structure.

Highlights

The highlights of Essential SAFe include the following:

Agile Release Train (ART) Δ a long-lived team of Agile teams that incrementally develops, delivers and often operates one or more solutions in a value stream.

Continuous Delivery Pipeline Δ Describes the workflows, activities, and automation needed to release value to the end user constantly.

Customer Centricity Δ A mindset and a way of doing business that focuses on creating positive customer experiences.

Design Thinking Δ An iterative solution development process that assures solutions are desired by customers and users while ensuring the solution is feasible, economically viable, and sustainable throughout its lifecycle.

Lean UX Δ Lean User Experience is a mindset, culture, and process that embraces Lean-Agile methods. It implements functionality in minimum viable increments and determines success by measuring results against a benefit hypothesis.

PI (Planning Interval) Δ is a fixed timebox during which an Agile Release (ART) Train delivers continuous value to achieve its PI Objectives.

Iterations Δ Fixed-length timeboxes that provide the development cadence for Agile teams building solutions.

Innovation and Planning (IP) Iteration Δ Provides the teams with an opportunity for exploration and innovation, dedicated time for planning, and learning through informal and formal channels.

SAFe Scrum Δ A lightweight process for Agile Teams to deliver value continuously. Most SAFe teams use SAFe Scrum to facilitate their work, including Built-in quality practices.

SAFe Team Kanban Δ a Lean-Agile method that helps Agile Teams facilitate the flow of value by visualizing workflow, establishing Work in Process (WIP) limits, delivering value continuously, measuring throughput, and continually improving their process.

Built-In Quality Δ a set of practices established to ensure that the outputs of business functions or technology teams meet appropriate quality standards throughout the process of creating customer value.

DevSecOps Δ a mindset, culture, and set of technical practices. It provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a system.

ART Flow Δ a state where an Agile Release Train (ART) delivers a continuous flow of value to the customer.

Team Flow Δ a state in which Agile Teams deliver a continuous flow of value to the customer.

Roles

ARTs are self-managing and self-organizing teams of Agile teams that plan, commit, and execute together. Agile Team and ART roles help guide and direct the ART, aligning them to a shared mission and providing the necessary Lean governance.

ART Roles

Product Management Ø responsible for defining and supporting the building of desirable, feasible, viable, and sustainable products that meet customer needs over the product-market lifecycle.

System Architect Ø an individual or small cross-discipline team that applies **Principle #2, Apply Systems Thinking**. They define the overall architecture of the system, help identify Nonfunctional Requirements (NFRs), determine the significant elements and subsystems, and help design the interfaces and collaborations among them.

Release Train Engineer (RTE) Ø a servant leader and the chief Scrum Master/Team Coach for the train. The RTE facilitates optimizing the flow of value by ensuring the ART events and artifacts function correctly, including the ART Kanban, Inspect & Adapt (I&A) workshop, ART Sync, and PI Planning.

Business Owners Ø a small group of stakeholders with the business and technical responsibility for fitness for use, governance, and return on investment (ROI) for a **Solution** developed by an ART. They are primary stakeholders in the ART and actively participate in ART events.

Team Roles

Agile Teams Ø a cross-functional group of ten or fewer individuals who can define, build, test, and deploy an increment of value in a short time box. Each ART comprises 5 Ø 15 Agile teams and includes the roles and infrastructure necessary to deliver fully working and tested business solutions.

Product Owner (PO) Ø the content authority for the team backlog is responsible for defining stories and prioritizing the backlog.

Scrum Master/Team Coach (SM/TC) Ø a servant leader and Agile team coach who helps the team remove impediments, facilitates team events, and fosters an environment for high-performing teams.

Events

Essential SAFe has multiple activities to help coordinate the ARTs and teams:

ART Events

PI Planning Ø a cadence-based, face-to-face planning event that serves as the heartbeat of the ART, aligning all the teams on the ART to the shared mission.

System Demo Ø provides an integrated view of new features from the most recent iteration delivered by all the teams in the ART. Each demo includes ART stakeholders with an objective measure of progress during a PI.

Inspect & Adapt Ø a significant event where the current state of the solution is demoed and evaluated. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop.

Coach Sync Ø helps coordinate the dependencies of the ARTs and provides visibility into progress and impediments.

[Product Owner \(PO\) Sync](#) ḏ provides visibility into how well the ART is progressing toward meeting the ART PI objectives, discusses problems or opportunities with feature development, and assesses any scope adjustments.

[ART Sync](#) ḏ combines the Coach Sync and PO Sync into a single event for an ART.

Team Events

SAFe Scrum teams and some SAFe Kanban teams manage their process with a series of regular events. These typically include the following:

[Iteration Planning](#) ḏ a team event in which an Agile team determines the iteration goals and how much of the team backlog they can commit to during an upcoming iteration. Team capacity determines the number of stories and enablers that are selected.

[Iteration Review](#) ḏ a cadence-based event at the end of each iteration in which the team reviews the previous increment's results and adjusts the team backlog based on feedback.

[Iteration Retrospective](#) ḏ an event held at the end of the iteration for the Agile team to review its practices and identify ways to improve. The retrospective applies qualitative and quantitative information presented during the iteration review.

[Backlog refinement](#) ḏ an event is held once or twice during the iteration to refine, review, and estimate future stories and enablers in the team backlog.

[Team Sync](#) ḏ a short meeting (usually 15 minutes or less), typically held about daily, to inspect progress toward the iteration goal, communicate, and adjust upcoming planned work.

Artifacts

The following Essential SAFe items help the ART coordinate and deliver work:

ART Artifacts

[Features](#) ḏ services that fulfill stakeholders' needs, sized to fit within the PI. Each includes a name, benefits hypothesis, and acceptance criteria.

[Enabler Features](#) ḏ supports the activities needed to extend the [Architectural Runway](#) to provide future business functionality, including exploration, architecture, infrastructure, and compliance.

[ART Epics](#) ḏ epics that a single ART can deliver.

[ART PI Objectives](#) ḏ describe the specific business and technical goals the ART intends to achieve in the upcoming PI.

[ART Backlog](#) ḏ a holding area for upcoming Features intended to address user needs and deliver business benefits for a single Agile Release Train (ART). It also contains the enabler features necessary to build the [Architectural Runway](#).

[Vision](#) ḏ describes the future state of the solutions under development. It reflects the customer and stakeholder needs and the features proposed to meet them.

Architectural Runway Ø consists of the existing code, components, and technical infrastructure necessary to implement prioritized, near-term features without excessive redesign and delay.

Solution Ø a product, service, or system ARTs deliver to the enterpriseÕs internal or external customers.

Solution Context Ø describes how the system will interface and be packaged and deployed in its operating environment.

Team Artifacts

Stories Ø provides short descriptions of a small piece of desired functionality written in the userÕs language.

Enabler stories provide the exploration, infrastructure, architecture, or compliance groundwork that another story or feature needs.

Team PI Objectives Ø summarized description of the specific business and technical goals an Agile team intends to achieve in the upcoming PI.

Team Backlog Ø consists of user and enabler stories; most are identified during PI planning and backlog refinement events.

Spanning Palette

Vision Ø describes a future view of the solution to be developed, reflecting customer and stakeholder needs and the Features and Capabilities proposed to address those needs.

Roadmap Ø communicates planned ART and value stream deliverables and milestones over a timeline.

System Team Ø a unique Agile team that assists in building and using the continuous delivery pipeline and, where necessary, validating full end-to-end system performance.

Ten Critical Success Factors

SAFe has proven to scale in all situations, from complex software and systems development to bond trading and medical devices to memory chips and fighter aircraft. But, with such a robust Framework, the question becomes: how closely does an organization need to follow various SAFe practices to get the desired result?

Also, when diagnosing SAFe implementation problems, it sometimes becomes apparent that enterprises may have skipped or stopped performing some of these critical practices. To support these challenges, the following Ten Critical Success Factors (Figure 2) highlight the minimal SAFe elements necessary for success.



Figure 2. Ten critical success factors

#1 ⚡ Lean-Agile Principles

SAFe practices are grounded in fundamental [Lean-Agile Principles](#). As organizations adopt SAFe, their continuous improvement activities find even better working methods. These principles guide those improvement efforts and ensure the adjustments move toward the Ÿshortest sustainable lead time, with the best quality and value to people and society.Ó

#2 ⚡ Real Agile Teams and Trains

Real [Agile Teams](#) and [ARTs](#) are fully cross-functional and can define, build and test their work. They have everything and everyone necessary to produce a working, tested increment of the solution. They are self-organizing and self-managing, enabling value to flow more quickly with minimal overhead.

#3 ⚡ Cadence and Synchronization

Cadence provides a rhythmic pattern, which offers a steady heartbeat for the development process. It makes routine those things that can be routine. Synchronization allows multiple perspectives to be understood and resolved at the same time. For example, synchronization pulls the various assets of a system together to assess solution-level viability.

#4 ⚡ PI Planning

No event is more powerful in SAFe than [PI planning](#) which provides the rhythm for the ART and connects strategy to execution by ensuring business and technology alignment. PI Planning is where *the people who do the work plan the work*. Aligning the entire ART with a common vision and goal creates energy and a shared

sense of purpose.

#5 ⚡ Customer Centricity, DevOps, and Release on Demand

SAFe enterprises create a positive [customer experience](#) across their products and services. They adopt a [DevOps](#) mindset, culture, and applicable technical practices to enable more frequent and higher-quality releases as [the market demands](#). These practices provide faster validation of hypotheses and produce greater profits, increased employee engagement, and more satisfied customers

#6 ⚡ System Demo

The primary measure of the ART's progress is the objective evidence provided by a working solution in the [System Demo](#). Every two weeks, the entire system—the integrated work of all teams on the train for that iteration—is demoed to the train's stakeholders. Stakeholders provide the feedback the train needs to stay on course and take corrective action. This feedback replaces other forms of governance that create additional work and slow flow.

#7 ⚡ Inspect and Adapt

[Inspect and Adapt](#) is a significant event held every PI. It is a regular time to reflect, collect data, and solve problems. The inspect and adapt event assembles teams and stakeholders to assess the solution and define improvements and actions needed to increase the following PI's velocity, quality, and reliability.

#8 ⚡ IP Iteration

The [Innovation and Planning Iteration](#) occurs every PI and serves multiple purposes. It is an estimating buffer for meeting PI objectives and provides dedicated time for innovation, continuing education, PI Planning, and Inspect and Adapt. IP Iteration activities enable many Lean-Agile principles that foster business agility.

#9 ⚡ Architectural Runway

[Architectural Runway](#) consists of the existing code, components, and technical infrastructure necessary to implement high-priority, near-term features without excessive delay and redesign. Insufficient investment in the architectural runway slows the train and makes the ART's delivery less predictable.

#10 ⚡ Lean-Agile Leadership

For SAFe to be effective, the enterprise's leaders and managers must take responsibility for Lean-Agile adoption and success. Executives and managers must become [Lean-Agile leaders](#) who are trained and then become trainers in these leaner ways of thinking and operating. Without leadership taking responsibility for the implementation, the transformation will likely fail to achieve the full benefits.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

Last update: 30 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

Become a Trainer

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*Develop on
Cadence. Release
on Demand.*

NA SAFe mantra

Release on Demand

Release on Demand is an aspect of the Continuous Delivery Pipeline that releases new functionality immediately or incrementally based on business and customer needs.

[Adjust Cookie Settings](#)

Release on Demand is the final aspect in the four-part [Continuous Delivery Pipeline](#)

of [Continuous Exploration](#) (CE), [Continuous Integration](#) (CI), [Continuous Deployment](#), and Release on Demand (Figure 1).

Figure 1. Release on demand is the final element of the continuous delivery pipeline

Since tangible development value only occurs when end users are operating the [Solution](#) in their environment, releasing that value at the right time is critical for the [Enterprise](#) to gain the real benefits of agility.

The decision of what and when to release is a critical economic driver that requires careful consideration. For many, *continuous delivery* is the desired end state, allowing the release of new functionality immediately after its deployment. But more often, the release is a decoupled, on-demand activity occurring for specific users, timed for when they need it or when it makes the most economic sense for customers and the business.

Details

The [Agile Product Delivery](#) competency article describes how the *Develop on cadence; release on demand* dimension creates the ability to deliver valuable solutions to end users with optimal timing and frequency. It raises the following three questions for [Product](#) and [Solution Management](#):

- When should a release happen?
- What elements of the solution should be released?
- Which end-users should receive the release?

A [Customer-Centric](#) mindset guides how Product and Solution Management answers these questions:

- Market rhythms and market events on the [Roadmap](#) inform release timing and aligns with customer needs
- Product Management should target release elements like [Features](#) or the entire system to specific customer segments

Decoupling releases provide additional benefits that promote [Business Agility](#), especially for [Operational Value Streams](#) serving external customers, for example:

- Product marketing can target promotional activities to specific audiences
- Sales teams can schedule activities with greater confidence in the timing and functionality of the solution

The Four Activities of Release on Demand

Figure 2 illustrates the four practices of releasing on demand:

1. **Release** Øthe practices needed to deliver the solution to end users, all at once or incrementally
2. **Stabilize and Operate** Ø ensures the solution is working well from a functional and [Nonfunctional requirements](#) (NFR) perspective
3. **Measure** Øhow to quantify if the newly-released functionality provides the intended value
4. **Learn** Øcollecting feedback and preparing for the next loop through the CDP

Figure 2. Four activities of release on demand

Figure 2. Four activities of release on demand

Release Value to Customers

When the **Solution** is in production and verified, it's time to make it available to customers. However, this timing is a critical business decision since releasing value too early or too late can adversely impact economics. In collaboration with other stakeholders, Product Management establishes policies that govern the release process, from automatically allowing qualified code to be immediately available to customers or holding a more formal review process with a manual gate. The more complex the system, the more likely there will be a manual gate to determine the answers to the earlier critical questions (what to release, to whom, and when).

The following practices contribute to the ability to release:

- Dark launches** □ Allows deployment to a production environment without releasing the functionality to end users
- Feature toggles** □ Provides a mechanism that allows code to be turned **on** or **off** without needing additional deployment.

Canary releases ☰ The practice of releasing the solution to a specific Customer segment and measuring the results before expanding and releasing it to more customers.

Decoupled release elements ☰ This technique identifies specific release elements, each of which can be released independently. Even simple solutions will have multiple release elements, each operating with different release strategies, as Figure 3 illustrates

Figure 3. Decouple release elements from the solution

Figure 3. Decouple release elements from the solution

For example, the SAFe website hosting this article has multiple and somewhat independent release cycles. Scaled Agile can:

Fix a security issue in our hosting infrastructure at any time (an ad hoc, but expedited, class of service)

Update any article at any time and notify readers via a blog post (high frequency)

Add new content to the [extended SAFe guidance](#) whenever it's available (medium frequency)

Create significant updates to the Framework, including a new Big Picture, at a frequency that balances the ability of customers to consume new versions and our development efforts (low frequency)

These separate flows – value streamlets – continue to represent a full, end-to-end flow of value within a Value Stream, each of which is managed to deliver value according to its own needs and pace. Identifying streamlets is critical to enable release on demand, as they allow the different elements of the solution to be released independently in a separate cadence. They also provide insights on the organization of teams and ARTs so that they can independently release on demand.

Stabilize and Operate

Once customers have access to newly verified and deployed solutions, unanticipated problems arise. Higher volumes of usage or unexpected usage patterns may cause these issues. Teams must quickly resolve incidents and security threats within their Service Level Agreements (SLAs). Several practices help operate the solution:

Cross-team collaboration – A mindset of cooperation across the [Value Stream](#) to identify and solve problems as they arise is crucial. This new thinking involves building Agile Release Trains to develop and *operate* the solution.

Failover/disaster recovery – Failures will occur. Developing a failover mechanism is vital to allow service to resume quickly or even avoid service interruption. Disaster recovery must be planned, architected into the service, and practiced.

Continuous security monitoring – Security as code and penetration testing focus on preventing known vulnerabilities from getting to production. But it's also essential to test services continuously for newly discovered and reported vulnerabilities and to detect intrusions and attacks on services and infrastructure.

Architect for operations – Enterprises must consider operational needs. High loads, security attacks, and responding to incidents motivate a range of options, from downgrading or removing services to adding capacity. Telemetry and logging capabilities enable organizations to understand, improve and tune their architecture to meet evolving usage patterns.

Monitor nonfunctional requirements (NFRs) – To avoid service disruptions, teams must continuously monitor system attributes such as reliability, performance, scalability, and more.

Measure the Business Value

The first activity of continuous exploration is to *hypothesize* and use application telemetry to measure if the hypothesis was proven and the business value delivered. Two practices support this effort:

Application telemetry ☰ Application telemetry is the primary mechanism to track and measure data usage against the hypothesis.

Innovation accounting ☰ Evaluating a hypothesis requires different metrics than those used to measure end-state working solutions. Innovation accounting measures the idea’s intermediate and predictive business outcomes during initial incremental solution development and evaluation of the Minimum Viable Product (MVP). (Read more in the [Innovation Accounting](#) article.)

Learn and React

The information gathered from the release is used to close the loop in the CDP. Product Management will use this feedback to make investment choices about Features and Epics. Part of the learning process is to analyze the information on how value flows to improve the CDP. Three practices help accomplish faster flow and higher value:

Lean startup thinking ☰ The benefit hypothesis for MVPs and MMFs is evaluated. If not proven, the organization decides if the development efforts should continue, stop, or pivot to a new idea and experiment with different approaches to achieve the strategy.

Value stream mapping ☰ An essential tool to improve the flow of value across the pipeline is value stream mapping. This tool provides the visibility needed to identify bottlenecks and problem areas to flow, design a future state, and create [Enablers](#) to improve the pipeline.

Relentless improvement ☰ The ART can continuously improve the flow of value. This mindset is part of the [SAFe Core Values](#) and is crucial to achieving results.

Customers realize value only when features are in their hands. When this value is measured, new knowledge informs the ongoing exploration efforts, starting the cycle anew. For one feature, this is the end of the pipeline. For another, however, it’s the beginning, as the continuous delivery process drives new value to users and new learning to the organization. Getting fast feedback is built into the process, allowing the teams to adjust to market needs.

Release Governance

Release governance is the process of planning, managing, and governing solution releases, which helps guide the value stream toward the business goals. In some enterprises, especially those with significant regulatory and compliance criteria, this is a centralized Portfolio SAFe team or function (*Release Management* is a common term) that ensures releases meet all the relevant business criteria.

In other circumstances, ART and Solution Train leadership and stakeholders from

development operations, quality, sales, and other stakeholders assume some release management and governance responsibilities.

In either case, release governance helps internal and external stakeholders receive and deploy the new solution, ensuring the ART addresses critical governance quality elements before deployment—*including internal and external security, regulatory, and other compliance concerns.*

ARTs plan releases during [PI Planning](#). That's the easy part. The difficulty lies in coordinating the implementation of features across multiple iterations in the [PI](#). This challenge is especially true when new issues, roadblocks, dependencies, and gaps in [Vision](#) and backlogs arise. Due to these challenges, the scope of each release must be continually managed, revalidated, and communicated. Primary considerations include:

- Ensuring the organization's release governance is understood and adhered
- Communicating release status to internal and external stakeholders
- Ensuring that an appropriate deployment plan is in place
- Coordinating with marketing and with Product and Solution Management on internal and external communications
- Validating that the solution meets relevant solution quality and [Compliance](#) criteria
- Participating in [Inspect and Adapt \(I&A\)](#) to improve the release process, value stream productivity, and solution quality
- Providing final authorization for the release
- Acting as a liaison with [Lean Portfolio Management \(LPM\)](#), as appropriate
- Participating in and overseeing the final release activities

Many enterprises hold *release governance* meetings on a regular cadence to address the following questions:

- Is the vision still understood, and are the trains and teams optimally aligned?
- Does everyone understand what they are building, and is their alignment with the [Value Stream](#)'s purpose and current [Strategic Themes](#)?
- Are the trains tracking to the desired release dates?
- Does the [Solution](#) have the appropriate built-in quality?
- What impediments must be resolved to facilitate progress?

The PO or ART Sync provides senior management with regular visibility into the release progress. It's also the place to approve any scope, timing, people, or resource adjustments necessary to ensure the release. In a more continuous delivery environment, the participants closely monitor the release section of the ART Kanban. They verify that items are released when needed to the right customer segments, manage canary and dark releases, evaluate hypotheses, and confirm the removal of feature toggles after production verification.

Enabling Release on Demand with DevOps

This aspect of the CDP reveals the cumulative value of all upstream efforts, closing the learning loop that began with continuous exploration. All activities must be quick, low-risk, aligned to business outcomes, and optimized for fast, accurate feedback. DevOps practices and tooling enable the responsiveness that is so critical in this “last mile” of the delivery pipeline.

Figure 4 illustrates how SAFe’s CALMR approach to DevOps (center) and several practice domains (inner rings) enable release on demand. Each of the four activities (in green) is a collaborative effort that draws upon DevOps expertise from multiple disciplines to maximize business value and validate learning.

Figure 4. DevOps enables release on demand

Figure 4. DevOps enables release on demand

Releasing, for example, requires immediate activation of deployed solutions using infrastructure configurations stored in version control, proactive monitoring that

informs operations teams of the health, security, and value of those solutions, and fast recovery from production issues specified in SLAs.

DevOps enables all four release on demand activities, though with different combinations of technical practices and tooling. See the [DevOps](#) article series for more guidance on how it empowers the CDP. Even after implementing a CDP and applying DevOps, organizations may still suffer delays that inhibit getting value to customers when business needs dictate. See the [ART Flow](#) article for more information on making value flow without interruption ([Principle #6](#)).

Learn More

[1] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Random House, Inc, 2011.

[2] Womack, Jim. *Gemba Walks Expanded 2nd Edition*. Lean Enterprise Institute, Inc, 2019.

[3] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (Agile Software Development Series)*. Pearson Education, 2011.

[4] <http://www.innovationgames.com>

[5] Gothelf, Jeff, and Josh Seiden. *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media, 2016.

Last update: 9 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



*Innovation comes
from the producer,
not the customer.*

ÑW. Edwards Deming

Portfolio Backlog

The Portfolio Backlog is a Kanban system that is used to capture and manage the business and enabler epics intended to create and evolve the portfolio's products, services, and solutions.

[Adjust Cookie Settings](#)

Details

Lean Portfolio Management (LPM) is responsible for developing, maintaining, and prioritizing the Portfolio backlog. They actively collaborate with stakeholders, including Business Owners (many of whom are part of LPM), Product and Solution Management, Epic Owners Enterprise Architects, and others, to discover the epics needed to advance the portfolio's solutions.

Portfolio Epics are large (and typically cross-cutting initiatives) managed through the Portfolio Kanban. Since more effort and capacity are required as the epic travels from the left to the right of the Kanban, LPM carefully decides which ones will proceed to each subsequent step or be removed altogether.

Building and Refining the Backlog

LPM applies a flow-based approach to build and refine the backlog, ensuring portfolio epics are ready for implementation with an appropriate level of discovery and risk.

Refining the portfolio backlog to ensure readiness often involves the following activities:

- Reviewing new epics and determining their alignment with the portfolio's strategic themes and vision

- Evaluating the Epic Hypothesis Statement to decide whether it warrants assignment to an Epic Owner

- Prioritizing the backlog using Weighted Shortest Job First (WSJF) and other factors in collaboration with Business Owners, Enterprise Architects, Product Management, and other stakeholders

Backlog refinement activities often occur during the Portfolio Sync and the Strategic Portfolio Review events. LPM and its stakeholders add new backlog items to the Funnel, update priorities, and remove less promising epics. A well-maintained backlog is a prerequisite for successful portfolio management.

Managing the Backlog with Kanban

The Portfolio Kanban system visualizes and facilitates the flow of business epics and enablers from idea through implementation.

Portfolio epics are made visible, developed, and managed through the Portfolio Kanban system, where they proceed through various process states until they are approved or rejected. Before being committed to implementation, epics require analysis and approval. Epic Owners take responsibility for the essential collaborations needed for this task, while Enterprise Architects typically guide the enabler epics that support the technical considerations for business epics.

Many companies struggle with innovation because they do not actively manage their portfolio with enough rigor or discipline, allowing less critical efforts to continue while starving or delaying those with more potential. Simply picking amongst the best big ideas will not advance the portfolio from its current state to a more promising and better future state. It requires a purposeful journey where LPM and its stakeholders identify specific epics to achieve the portfolio's [Vision](#).

The portfolio Kanban helps align strategy and execution by identifying, communicating, and governing the selection of the largest and most strategic initiatives (epics) for a SAFe portfolio. LPM is responsible for the Kanban portfolio, which is often used during the strategic portfolio review and portfolio sync events to manage and monitor the flow of epics.

While opportunistic epics are welcome, LPM has a fiduciary responsibility to facilitate the flow of epics best aligned with the portfolio's strategic themes, vision, and economic framework. Achieving a better future state requires the [Development Value Streams](#) to adopt a "one-portfolio" mindset, where they cooperate to achieve the portfolio's higher-level objectives and the broader aim of the enterprise.

Figure 1 illustrates an epic's process from the *Funnel* to completion, where the Epic Owner coordinates its advancement through the portfolio Kanban. Only the *Reviewing, Analyzing, and Ready* states have explicit WIP limits. The *MVP* sub-state has an implicit WIP limit, the available capacity of trains, and the value streams' budget.

Figure 1. Example portfolio Kanban system and its collaborators

Figure 1. Example portfolio Kanban system and its collaborators

The Kanban^{Os} design should evolve to match the needs of a specific portfolio and reflect continuous improvements to the process. These improvements may include adjusting WIP limits, splitting or combining Kanban states, or adding classes of service to optimize the flow and priority of epics.

Each example portfolio state is described further in the following sections.

Funnel

The Funnel is used to intake all significant business and technology ideas for a specific portfolio. These ideas may originate as strategic concerns, arise from Agile Teams or ARTs, or suggestions from customers and partners. These ideas are anticipated to be large enough to exceed the epic threshold [Guardrail](#) or perhaps have some other strategic or business model impact.

Epics are initially described in the Epic Hypothesis Statement (see [Epic](#) article). The Funnel is not WIP limited as these epics are simply ideas that may deserve additional consideration. If an initial review of an idea is not likely to exceed the epic threshold guardrail or be a portfolio concern, it moves to the ART or Solution Train Kanban. While they can arise from any source, Figure 2 illustrates how epics typically flow into the Funnel:

Maintaining the [Portfolio Vision](#) and [Roadmap](#) identifies new initiatives that feed directly into the portfolio Kanban. These initiatives may include updates from the Enterprise strategy or supporting other portfolios.

The [Continuous Exploration](#) process discovers user and market needs and may result in identifying epics.

Figure 2. Feeding the portfolio Funnel

Figure 2. Feeding the portfolio Funnel

Reviewing

Since epics are some of the most significant portfolio investments, an Epic Owner is

needed to sponsor the epic and define its intent. When an Epic Owner is available, they pull the epic into this state, working with relevant stakeholders to refine and further elaborate the Epic Hypothesis Statement.

A WIP limit for this state is typically specified. Moreover, the lack of an Epic Owner available to do the work can serve as an implicit WIP limit. A WSJF estimate relative to other items in the reviewing state is established based on preliminary sizing and cost estimates. Suppose the epic does not appear sufficiently viable or aligned with the portfolio's strategic themes or other guardrails. In that case, it moves to the Done state, which frees capacity for more promising alternatives.

Analyzing

When the Epic Owner has the capacity and room within the WIP limit, epics with high potential are pulled into the Analyzing state. Epics that make it here deserve more rigorous analysis and further investment. This state typically requires active collaboration among the following roles:

Business Owners

Enterprise Architects and [System](#) and [Solution Architects](#)

[Product](#) and [Solution Management](#)

[Agile Teams](#)

During the analysis state, the following activities typically occur:

Identification and review of solution alternatives

Defining the MVP

Establishing refined cost estimates for the MVP and its full anticipated scope

Creating the Lean Business Case

Small research spikes to demonstrate potential technical and business viability

Initial customer validation

Updating WSJF relative to other epics in this state

Go/No-go decision by LPM based on the Lean Business Case

Typically, there are only a small number of epics in this state, and they are reviewed routinely by LPM. Since the epic's analysis and eventual implementation consume precious capacity, approval to move into the next state is more rigorous. WSJF is one factor, but many additional considerations, such as the Lean Business Case, may also be applied.

LPM uses the Lean Business Case during the portfolio sync to make a "Go/No-go" decision. "Go" confirms the epic is approved for implementation and sequenced using WSJF. "No-go" moves the epic to done.

Ready

Epics in the analyzing state with the highest WSJF are pulled into the next state, Ready, as soon as space is available. This state is a low-cost wait state where epics are periodically reviewed and prioritized by updating WSJF and other relevant factors.

Implementing

Since many companies will generate more good ideas than they can fund, LPM and participants from different [Value Streams](#) collaboratively determine which epics should be implemented based on [Participatory Budgeting \(PB\)](#) or a similar process. The results of the PB event provide feedback to help LPM decide how to adjust value stream budgets to support the implementation of the most beneficial business and enabler epics.

The implementation of epics also includes considerations for sequencing work, sizing, and ranking the epics relative to each other, typically by one final [Weighted Shortest Job First \(WSJF\)](#) prioritization. Adjustments to epic priority from WSJF may be necessary to stay within the Lean budget guardrails, such as capacity allocation, investment horizons, and other factors. LPM should have a good understanding of available ART capacity since the job duration used by WSJF may be affected by implementation capacity.

The [Epic Owner](#) works with various stakeholders to split epics into [Features](#) and [Capabilities](#). The responsible ARTs and Solution Trains pull the work into their respective backlogs, as illustrated in Figure 3.

Figure 3. ARTs and Solution Trains pull work into their backlogs

Figure 3. ARTs and Solution Trains pull work into their backlogs

When ready for implementation, features and capabilities are presented at the relevant [PI Planning](#) events to begin the development of the epic's MVP. LPM evaluates the MVP's progress based on leading indicators, Value Stream [KPIs](#), and demos.

The implementation step has two sub-states, **MVP** and **Persevere**. They are described in the following sections:

MVP

When sufficient capacity from one or more ARTs is available, the epics with the highest WSJF advance to the MVP state. Here, the Epic Owner works with the Agile Teams to begin the activities needed to develop the MVP and evaluate its business outcome hypothesis. Work on the MVP continues until the money allocated for the MVP has been spent or the hypothesis can be evaluated.

If the value stream runs out of money to implement the MVP and the need for the

epic still exists, a new one can be proposed and placed in the Funnel. The original epic moves to **Done**. Of course, LPM can adapt this rule (and any others) to meet the needs of their organization.

Persevere

If the hypothesis is proven true, the epic advances to the Persevere state, and teams will continue to implement additional features and capabilities. ARTs manage the additional investment via ongoing feature prioritization in the [ART Backlog](#) in various value streams. Eventually, the epic will be “done enough” such that ongoing WSJF will prioritize new capabilities and features from other sources as a higher priority. Epic Owners remain available to assist ARTs and Solution Trains responsible for implementation.

Done

An epic is considered Done when sufficient knowledge or value is achieved, or it is no longer a portfolio concern. Completing the fully envisioned scope from the Lean business case is not a criterion. Instead, the epic is **Done** if:

- It is ejected from the portfolio Kanban by LPM in any of the earlier states
- The hypothesis is proven, and LPM has determined that additional portfolio governance is no longer required

If the hypothesis is proven, work on the epic may continue by various ARTs responsible for its implementation. The Epic Owner may need to provide ongoing guidance and follow-up. Since the epic is no longer a portfolio concern, leading indicators, value stream [KPIs](#), and [Guardrails](#) are used to keep LPM informed of progress.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Anderson, David. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The greater the potential for reward in the value portfolio, the less risk there is.

~Warren Buffet

Portfolio SAFe

Portfolio SAFe provides strategy and investment funding, Agile portfolio operations, and Lean governance for one or more value streams.

[Adjust Cookie Settings](#)

Portfolio SAFe (Figure 1) helps establish strategy and investment funding, Agile

portfolio operations, and Lean governance for the value streams that constitute a portfolio. It describes the roles, events, and artifacts to help those accountable for the overall business outcomes and the challenge of defining, communicating, and aligning strategy with execution. It includes the following constructs:

- The [Essential SAFe](#) configuration
- Two additional Competences, [Lean Portfolio Management](#) and [Organizational Agility](#)
- The Portfolio-level roles, artifacts, and events
- The full spanning palette
- A connection to the [Enterprise](#) or [Government](#) entity that the portfolio is a part of

Figure 1. Portfolio SAFe Configuration

Figure 1. Portfolio SAFe Configuration

Details

A SAFe portfolio aligns strategy to execution via a collection of [Development Value Streams](#). Each develops one or more [Solutions](#) the portfolio needs to accomplish its business mission and vision, operating under a shared governance model.

The portfolio is a collection of related development value streams that organize [Agile Teams](#) and [Agile Release Trains](#) around the solutions needed for a particular business area. It provides essential funding and minimum governance, including [Lean Budget Guardrails](#) that facilitate decentralized decision-making. The portfolio helps ensure value streams focus on building the right things with the appropriate level of investment in solutions to meet the portfolio's strategic objectives.

The remainder of this article describes the highlights, roles, events, and artifacts present at the portfolio level (Figure 2).

Figure 2. Portfolio level elements

Figure 2. Portfolio level elements

Portfolio Level Highlights

Highlights of the Portfolio level include:

[Portfolio Flow](#) □ Describes how Lean Portfolio Management (LPM) provides a continuous flow of [Epics](#) to achieve the portfolio's vision and enterprise business

objectives.

Strategic Themes are portfolio-level business objectives that provide competitive differentiation and strategic advantage.

Portfolio Vision ▷ The portfolio vision represents the future desired state of a portfolio's **Value Streams** and solutions. It describes how they will cooperate to achieve their objectives and the broader aim of the Enterprise.

Portfolio Backlog ▷ The Portfolio backlog contains business and enabler Epics needed to create and evolve the portfolio's products, services, and systems that typically cross multiple value streams and PIs. These epics are visualized and managed through the Portfolio Kanban, where they proceed through various process states.

Big Data ▷ SAFe addresses big data concerns at the portfolio level as it requires vision, investment, some degree of centralization, and governance within and across the value streams within the portfolio.

Lean Budgets ▷ Lean budgeting allows fast and empowered decision-making, with appropriate financial control and accountability through **Guardrails**.

Participatory Budgeting ▷ A collaborative process for allocating the portfolio budget to its value streams.

Value Stream Management ▷ Value Stream Management (VSM) is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle.

Coordination ▷ Value Stream Coordination defines how to manage dependencies and exploit the opportunities that exist in the interconnections between value streams.

Development Value Streams ▷ Every development value stream funds the people and resources needed to build **Solutions** that deliver value to an internal or external customer.

Solutions ▷ The portfolio's products, systems, or services that the development value streams build and maintain for their related **Operational Value Streams**.

KPIs ▷ Value Stream Key Performance Indicators (KPIs) are the quantifiable measures used to evaluate how a value stream performs against its business objectives.

Roles

The following roles provide the highest level of accountability and governance within a portfolio, including coordinating multiple value streams.

Lean Portfolio Management (LPM) is a function fulfilled by individuals with decision-making and financial accountability for a SAFe portfolio.

Epic Owners collaboratively define the epic, including a Minimum Viable Product (MVP) and Lean business case, and facilitate its implementation when approved.

Enterprise Architects provide the strategic technical direction and roadmap, enabling a portfolio to support the current and future business capabilities.

Events

The following events support the alignment of strategy with execution:

[Portfolio Sync](#) provides visibility into how well the portfolio is progressing toward meeting its strategic objectives. It typically includes reviewing the value stream and ART execution and governance of other portfolio investments.

[Participatory Budgeting](#) enables LPM to collaborate with [Business Owners](#) and other relevant stakeholders to right-size the investments in value streams. It helps manage the approval process of epics in the portfolio Kanban.

[Strategic Portfolio Review](#) enables LPM to create alignment and investment guidance to inform rapid, high-quality, decentralized decisions, adapt to meet changing needs, and provide governance to respond effectively to new and changing opportunities.

Artifacts

The following artifacts help describe the strategic intent of the portfolio solution set:

[Strategic Themes](#) are portfolio-level business objectives that provide competitive differentiation and strategic advantage.

[Portfolio Canvas](#) defines the portfolio's value propositions, key resources and activities, cost structure, and revenue streams.

[Guardrails](#) describe the portfolio's policies and practices for budgeting, spending, and governance.

[Portfolio Epics](#) represent the more substantial investments within a portfolio. These large initiatives typically cross multiple value streams and [PIs](#). There are two types of portfolio epics: [Business Epics](#) and [Enabler Epics](#).

[Nonfunctional Requirements \(NFRs\)](#) define system attributes such as security, reliability, performance, maintainability, scalability, and usability and serve as constraints or restrictions on the system's design

[Portfolio Backlog](#) contains the business and enabler epics needed to create and evolve the portfolio's products, services, and systems. The Portfolio Kanban system manages the portfolio backlog.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled: Achieving Business*

Agility with the Scaled Agile Framework. Addison-Wesley, 2020.

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Luck is what happens when preparation meets opportunity.

ÑSeneca

Enablers

Enablers are backlog items that extend the architectural runway of the solution under development or improve the performance of the development value stream.

Adjust Cookie Settings

Enablers are captured in backlogs as a type of [Epic](#), [Capability](#), [Feature](#), or [Story](#). They

are used primarily for exploration, architecture implementation, refactoring, building infrastructure, and addressing compliance. While their type is unique, they are managed similarly to customer-facing backlog items.

Details

Enablers bring visibility to all the work necessary to support the efficient development and delivery of future business requirements. Enablers are used to explore ideas, improve architecture, strengthen infrastructure, and manage compliance. Since enablers result in the production of tangible outputs, they must be visible. They are treated like all other backlog items—subject to visibility, prioritization, incremental delivery, measurement, and feedback.

Types of Enablers

Enablers can be used to define any activity that improves the value stream in support of foreseeable business needs. These activities generally fall into one of four categories:

Exploration ▷ support research, prototyping, and other activities needed to develop an understanding of customer needs, including the exploration of prospective [Solutions](#) and evaluation of alternatives

Architectural ▷ used to build [Architectural Runway](#), which allows smoother and faster development through the [Continuous Delivery Pipeline](#) (CDP)

Infrastructure ▷ support the creation and optimization of the development and runtime environments that host the systems used to build, validate, deploy, and operate solutions

Compliance ▷ facilitate managing specific compliance activities, including Verification and Validation (V&V), audits and approvals, and policy automation

Creating and Managing Enablers

Enablers exist throughout SAFe and are written and prioritized according to the same rules as their corresponding epics, features, capabilities, and stories.

Enabler Epics ▷ These are written using the “epic hypothesis statement” format, in the same way as business epics. Enabler epics can span multiple [Agile Release Trains](#) (ARTs) and [PIs](#) and are managed via the [Portfolio Backlog](#) and associated Kanban system.

Enabler Features and Capabilities ▷ These are defined by ARTs and [Solution Trains](#) and include a short phrase, benefit hypothesis, and acceptance criteria. They

must be sized to fit within a single PI.

Enabler Stories ☰ Must fit within [Iterations](#) like any story. Although they may not require the user voice format, their acceptance criteria clarify the requirements and support testing.

Architects often define and guide enabler epics, features, and capabilities. They might be [Enterprise Architects](#) supporting the portfolio, [System Architects](#) supporting ARTs, or [Solution Architects](#) supporting [Solution Trains](#). Architects steer enablers through the appropriate Kanban system and backlog, guiding implementation from concept to delivery. [Agile Teams](#) also use enablers; enabler stories emerge locally from their needs and are carried in the [Team Backlog](#).

The following examples illustrate how Agile teams and architects create and manage each of the four enabler types.

Enabling Exploration

Exploration enablers provide work items teams can use to discover requirements and design details. The nature of [Solution Intent](#) is that many requirements begin as variable intent. At the beginning of development, little is known about what the customer needs or how to implement it. Customers themselves often don't understand precisely what they need. Through [Continuous Exploration](#), teams progressively learn which aspects of solution intent should move from variable to fixed.

In an even broader view, there are typically many technical possibilities for implementing an identified business need or opportunity. Those alternatives must be analyzed and are often evaluated through modeling, prototyping, Set-Based Design, or the Lean Startup Cycle. Exploration enablers formalize these activities, assure that the work is visible, and help ensure solution development is closely aligned with the needs of customers and stakeholders.

Enabling Architecture

In SAFe, [Agile Architecture](#) practices produce architectural runway, the underlying technology that enables Agile teams and ARTs to deliver business solutions quickly. But the runway is constantly consumed by business epics, features, capabilities, and stories, so it must be extended for new functionality. Architectural enablers are used to build, extend, and maintain the runway.

Architectural enablers can also address problems with the resiliency of deployed solutions. After implementation, these enablers often reflect [Nonfunctional](#)

[Requirements](#) (NFRs) imposed on future backlog items. NFRs often originate as architectural enablers and grow as a set over time (Figure 1).

Figure 1. Many NFRs appear over time as a result of enablers

Figure 1. Many NFRs appear over time as a result of enablers

Enabling Infrastructure

Agile development requires frequent integration. Agile teams integrate their work and showcase the working solution increment at the [System Demo](#). Similarly, ARTs that are part of a Solution Train integrate their work as frequently as possible during the PI in preparation for [Solution Demos](#). Infrastructure enablers provide the [Continuous Integration](#) and [Continuous Deployment](#) technology that supports this aggressive integration cadence.

The [System Team](#) is integral in defining and building infrastructure enablers that enhance the development environment and streamline the CDP. [Shared Services](#), Operations teams, and Site Reliability Engineering (SRE) leverage infrastructure enablers to deliver [Cloud](#) services that accelerate solution development and solution scalability.

Enabling Compliance

By incrementally building the necessary artifacts in the solution intent over a series of PIs, SAFe supports continuous verification and validation. Verification activities are conducted as part of the development workflow and are often enforced in the Definition of Done (DoD). While the artifacts will satisfy the objective evidence needed at the end of development, they are created iteratively throughout the life cycle. Validation occurs when Product Owners, customers, and end-users participate in ART planning and system demos, validating fitness for purpose.

Enablers are used to support these activities. For example, consider a regulation that requires design reviews and that all actions stemming from those reviews must be documented when completed. A "design review enabler" backlog item would offer evidence of the review, and its DoD would ensure that actions are recorded and resolved according to the Lean Quality Management System (QMS). If needed, the activities themselves could be tracked as enabler stories.

Implementing Enablers Incrementally

Enablers supply critical, foundational technology and insight into the value stream. Consequently, they deserve focused attention in the portfolio, from budgeting and capacity allocation through delivery and ongoing improvement. But because the actual value of enablers is tied to the realization of future business objectives, care must be taken to implement enablers quickly and iteratively. Otherwise, delivering value to customers can be significantly delayed, undermining the fundamental purpose of enablers.

Enablers of all types should be implemented incrementally. However, because architectural and infrastructure enablers often influence the delivery and operation of mission-critical solutions, they deserve special mention here.

The size and demands of architectural and infrastructure enabler work can be overwhelming. So, it's important to remember that they need to be split into features and stories that can be delivered incrementally. This can be difficult, however, as architectural and infrastructure changes can potentially stop the existing system from working until the changes are in place.

The work must be sequenced to ensure the system can continue operating in the current environment while enablers are implemented. That way, teams can continue to work, integrate, demo, and even release new functionality.

There are three ways to approach this [1]:

Case A The enabler is big, but there is an incremental approach to

implementation. The system always runs (operates).

Case B ☰ The enabler is big but cannot be implemented incrementally. The system will need to take an occasional break.

Case C ☰ The enabler is *really* big, and it cannot be implemented incrementally. The system runs when needed.

Examples of incremental patterns are also described in [2], where the legacy subsystems are gradually “strangled” over time, using proven patterns such as asset capture or event interception.

By creating the technology platforms that deliver business functionality, enablers drive better economics. But innovative product development cannot occur without risk-taking. Therefore, initial technology-related decisions cannot always be correct, which is why the Lean enterprise must be prepared to change course occasionally. In these cases, the principle of sunk costs [3] provides essential guidance: Do not consider money already spent. Implementing incrementally allows corrective action before the investment grows too large.

Implementing Enablers Across ARTs and Value Streams

Enabler epics and capabilities can cut across multiple value streams or ARTs. During the analysis stage of the appropriate Kanban system, it is important to determine whether to implement the enabler across all ARTs simultaneously or incrementally (Figure 2).

Figure 2. Two approaches to implementing cross-cutting enablers

Figure 2. Two approaches to implementing cross-cutting enablers

In scenario A, the enabler is implemented first in ART 1, then by the other ARTs in subsequent PIs. This may lessen the impact of the change across the portfolio but can delay the full benefits of a fully implemented enabler. By contrast, scenario B calls for all ARTs to implement the enabler at the same time. This is preferable if the cost of delaying the entire implementation is unacceptably high.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Fowler, Martin. *Strangler Application*.
<http://martinfowler.com/bliki/StranglerApplication.html>

[3] Reinertsen, Donald. *The Principles of Product Development Flow: Second*

Generation Lean Product Development. Celeritas Publishing, 2009.

Last update: 13 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



What if we found ourselves building something that nobody wanted? In that case, what did it matter if we did it on time and on budget?

~Eric Ries

Epic

An Epic is a significant solution development initiative.

[Adjust Cookie Settings](#)

Due to their considerable scope and impact, epics require the definition of a

Minimum Viable Product (MVP) and approval by Lean Portfolio Management (LPM).

Portfolio epics are typically cross-cutting, typically spanning multiple [Value Streams](#) and [PIs](#). To accelerate learning and development and reduce risk, SAFe recommends applying the Lean Startup *build-measure-learn* cycle for these epics.

Details

This article describes the portfolio epic's definition, approval, and implementation. [Agile Release Train \(ART\)](#) and [Solution Train](#) epics, which follow a similar pattern, are described briefly at the end of this article.

There are two types of epics, each of which may occur at different levels of the Framework. Business epics directly deliver business value, while enabler epics advance the [Architectural Runway](#) to support upcoming business or technical needs.

It's important to note that epics are not merely a synonym for projects; they operate quite differently, as Figure 1 highlights.

Figure 1. Epics are not projects

Figure 1. Epics are not projects

SAFe discourages using the project funding model (refer to the [Lean Portfolio Management](#) article). Instead, the funding to implement epics is allocated directly to the value streams within a portfolio. Moreover, Agile Release Trains (ARTs) develop and deliver epics following the Lean Startup Cycle discussed later in this article (Figure 6).

Defining Epics

Since epics are some of the most significant enterprise investments, stakeholders must agree on their intent and definition. Figure 2 provides an epic hypothesis statement template for capturing, organizing, and communicating critical information about an epic.

Figure 2. Epic hypothesis statement

Figure 2. Epic hypothesis statement

[Download](#)

Epics above the approval [Guardrail](#) are made visible, developed, and managed through the [Portfolio Kanban](#) system, where they proceed through various states of maturity until they're approved or rejected. Before being committed to implementation, epics require analysis. [Epic Owners](#) take responsibility for the critical collaborations needed for Business Epics, while [Enterprise Architects](#) typically guide the [Enabler](#) epics that support the technical considerations for business epics.

Creating the Lean Business Case

The result of the epic analysis is a Lean business case (Figure 3).



Figure 3. Lean business case

[Download](#)

The LPM reviews the Lean business case to make a go/no-go decision for the epic. Once approved, portfolio epics move to the *Ready* state of the Portfolio Kanban. When capacity and budget become available from one or more ARTs, the Epic is pulled into implementation. The Epic Owner is responsible for working with [Product](#) and [Solution Management](#) and [System Architects](#) to split the epic into Features or Capabilities during backlog refinement. Epic Owners help prioritize these items in their respective backlogs and have ongoing responsibilities for their development and follow-up.

Defining the MVP

Analysis of an epic includes the definition of a Minimum Viable Product (MVP) for the epic. In the context of SAFe, an MVP is an early and minimal version of a new product or business [Solution](#) used to prove or disprove the epic hypothesis. Unlike

storyboards, prototypes, mockups, wireframes, and other exploratory techniques, the MVP is an actual product that real customers can use to generate validated learning.

Estimating Epic Costs

As Epics progress through the Portfolio Kanban, the LPM team will eventually need to understand the potential investment required to realize the hypothesized value. This analysis requires a meaningful estimate of the cost of the MVP, and the forecasted cost of the full implementation should the epic hypothesis be proven true.

The **MVP cost** ensures the portfolio is budgeting enough money to prove or disprove the Epic hypothesis. It helps ensure that LPM makes sufficient investments in innovation aligned with Lean budget guardrails.

The **forecasted implementation cost** considers ROI analysis, helping determine if the business case is sound, and allows the LPM team to prepare for potential adjustments to value stream budgets.

The Epic owner determines the amount of the MVP's investment in collaboration with other key stakeholders. This investment should be sufficient to prove or disprove the MVP's hypothesis. Once approved, the value stream cannot spend more than the defined investment cost to build and evaluate the MVP. If the value stream has evidence that this cost will be exceeded during epic implementation, further work on the epic should be discussed with LPM before exceeding the MVP's estimated cost.

Estimating Implementation Cost

Considerable strategic efforts often require collaboration with external Suppliers to develop Solutions. The MVP and the anticipated full implementation cost estimates should include internal costs and forecasted external Supplier expenses.

Estimating epics in the early stages can be challenging since there is limited data and learning at this point. As illustrated in Figure 4, "T-shirt sizing" is a simple way to estimate epics, especially in the early stages:

- A cost range is established for each t-shirt size using historical data
- The gaps in the cost ranges reflect the uncertainty of estimates and avoid excessive discussion around edge cases
- Each portfolio must determine the relevant cost range for the t-shirt sizes

The Epic Owner can incrementally refine the total implementation cost as the MVP is built and learning occurs.

Figure 4. Estimating epics using example T-shirt sizes and an example cost range in one enterprise

Supplier Costs

An epic investment often includes the contribution and cost from suppliers, whether internal or external. Ideally, enterprises engage external suppliers via Agile contracts, which supports estimating the costs of a supplier's contribution to a specific epic. For more on this topic, see the [Agile Contracts](#) extended guidance article.

Forecasting an Epic's Duration

While it can be challenging to forecast the duration of an epic implemented by a mix of internal ARTs and external suppliers, an understanding of the forecasted duration of the epic is critical to the proper functioning of the portfolio.

Like an epic's cost, its duration isn't easy to forecast as it includes several components, such as internal duration, supplier duration, and the collaborations

and interactions between the internal and external teams. Practically, unless the epic is wholly outsourced, LPM can focus on forecasts of the internal ARTs affected by the epic, as they are expected to coordinate work with external suppliers.

Forecasting an epic's duration requires an understanding of three data points:

1. An epic's estimated size in story points for each affected ART can also be calculated using T-shirt sizes and replacing the cost range with a story point range.
2. The historical velocity of the impacted ARTs.
3. The percent (%) capacity allocation that ARTs can dedicate to working on the epic.
This allocation typically results from negotiation between Product and Solution Management, Epic Owners, and LPM.

In the example shown in Figure 5, a portfolio has a substantial enabler epic that affects three ARTs, and LPM seeks to gain an estimate of the forecasted number of PIs.

ART 1 has estimated the epic's size as 2,000 ÷ 2,500 points. Product Management determines that ART 1 can allocate 40% of total capacity toward implementing its part of the epic. With a historical velocity of 1,000 story points per PI, ART 1 forecasts between five to seven PIs for the epic.

Figure 5. Example worksheet for forecasting an epic's duration

Figure 5. Example worksheet for forecasting an epic's duration

After repeating these calculations for each ART, the Epic Owner can see that some ARTs will likely be ready to release on demand earlier than others. However, the forecasted duration to deliver the entire epic across all ARTs will likely be between six and eight PIs. If this forecast does not align with business needs, negotiations such as adjusting capacity allocations or increasing the budget for suppliers will ensue. The Epic Owner updates the forecasted completion once work begins on the epic.

Implementing Epics

The SAFe Lean startup strategy recommends a highly iterative build-measure-learn cycle for product innovation and strategic investments. This approach for implementing epics provides the economic and strategic advantages of a Lean startup by managing investment and risk incrementally while leveraging the flow and visibility benefits of SAFe (Figure 6).

Gathering the data necessary to prove or disprove the epic hypothesis is highly iterative. These iterations continue until a data-driven result is obtained or the teams consume the entirety of the MVP budget. In general, the result of a proven hypothesis is an MVP suitable for continued investment by the value streams. Otherwise, any further investment requires the creation of a new epic.

Figure 6. Epics in the Lean startup cycle

Figure 6. Epics in the Lean startup cycle

After it's approved for implementation, the Epic Owner works with the [Agile Teams](#) to begin the development activities needed to realize the business outcomes hypothesis for the epic:

1. **If the hypothesis is true**, the epic enters the persevere state, which will drive more work by implementing additional features and capabilities. ARTs manage any further investment in the epic via ongoing WSJF feature prioritization of the [ART Backlog](#). Local features identified by the ART, and those from the epic, compete during routine WSJF reprioritization.
2. **If the hypothesis is false**, Epic Owners can decide to pivot by creating a new epic

for LPM review or dropping the initiative altogether and switching to other work in the backlog.

After evaluating an epic's hypothesis, it may or may not be considered a portfolio concern. However, the Epic Owner may have ongoing stewardship and follow-up responsibilities.

Lean budgets' empowerment and decentralized decision-making depend on [Guardrails](#) for specific checks and balances. Value stream KPIs and other metrics also support guardrails to keep the LPM informed of the epic's progress toward meeting its business outcomes hypothesis.

ART and Solution Train Epics

Epics may also originate from local ARTs or Solution Trains, often starting as initiatives that warrant LPM attention because of their significant business impact or initiatives that exceed the epic threshold. These epics deserve a Lean business case and review and approval through the Portfolio Kanban system. The [ART and Solution Train Backlogs](#) article describes methods for managing the flow of local epics which do not meet the criteria for portfolio attention.

Learn More

[1] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Random House, Inc, 2011.

Last update: 9 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)

[Watch and download](#)
SAFe videos and
presentations

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use](#)
SAFe content and
trademarks
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



The devil is in the details.

—Common proverb

Nonfunctional Requirements

Nonfunctional Requirements (NFRs) are system qualities that guide the design of the solution and often serve as constraints across the relevant backlogs.

As opposed to functional requirements, which specify how a system responds to specific inputs, nonfunctional requirements are used to specify various system qualities and attributes, such as:

Adjust Cookies Settings

Performance: How fast a system should respond to requests

Scalability: How well a system can handle an increase in users or workload

Security: How well a system protects against unauthorized access and data breaches

Usability: How easy a system is to use

Maintainability: How easy it is to update and modify the system

NFRs are persistent qualities and constraints typically revisited as part of the definition of done (DoD) for each [Iteration](#), [PI](#), or [release](#). NFRs influence [Teams](#), [ART](#), [Solution Train](#), and [Portfolio](#) backlogs.

Details

Nonfunctional Requirements (NFRs) are intended to specify Ôsystem qualities,Õ various systems attributes that are not directly related to their functionality. These attributes do not tell what the system does but how well it does it. In contrast, *functional requirements* are expressed as [Capabilities](#), [Features](#), and [Stories](#), which define what the system does in response to various inputs. Although they may seem subtle, NFRs are just as vital to system success. Failing to meet NFRs can result in systems that do not meet the needs of the business, Customers, market, or applicable regulations or standards. In some cases, non-compliance can cause significant issues, such as cost, recalls, privacy, security, safety risk, legal exposure, and more.

Proper identification and implementation of NFRs are critical. The solution may be too costly or unviable if they are over-specified. If underspecified, the system may be inadequate for its intended use. No matter the scope of the system, an adaptive and incremental approach to exploring, defining, and implementing NFRs is a vital skill for Agile teams.

NFRs Constrain Backlogs

NFRs are associated with backlogs throughout SAFe, as Figure 1 illustrates. However, they are not backlog items. Backlog items come and go as they are implemented. NFRs are persistent constraints on the design and development of the system. For example, consider a requirement such as Ôall products in the suite require SAML-based single sign-on. While single sign-on is a functional requirement, the selection of SAML (Security Assertion Markup Language) is a constraint. Any new backlog item requiring sign-on functionality must include SAML in its acceptance criteria.

Figure 1. NFRs occur with all backlogs in SAFe

Because NFRs are significant attributes of the solution that the [Agile Release Train \(ART\)](#) and [Value Streams](#) create, they influence the backlogs of Teams, ARTs, and [Solution Trains](#). The portfolio backlog may also require NFRs, typically for cross-solution qualities (like regulatory standards). [Agile Teams](#) use [Built-In Quality](#) practices to accelerate NFR testing and make it continuous. Teams include the relevant NFRs in their DoD, use them as constraints on local design and implementation decisions, and take responsibility for NFR testing.

As illustrated in Figure 3, NFRs are modeled as backlog constraints.

Figure 2. NFRs are associated with backlogs and constrain the design of the system

NFRs may constrain any backlog items as described in the [SAFe Requirements Model](#). Most NFRs require one or more system quality tests (ideally automated) to know that the system complies with the constraint.

Types of NFRs

Generally speaking, there are two types of NFRs: *Systems Qualities* and *Design Constraints*. Each is described in the following sections.

System Qualities

NFRs are often architecturally significant requirements that describe the system's various quality attributes (ilities). They are as critical or even more critical than functional requirements that pass through the backlog. Working with [Product](#) and [Solution](#) Management and the teams, [System](#) and [Solution Architects](#) are often

responsible for identifying and establishing NFRs. Figure 3 shows a relatively comprehensive list of NFR sources to consider during development.

Figure 3. Examples of system attributes that may be subject to NFRs

Figure 3. Examples of system attributes that may be subject to NFRs [1]

Design Constraints

In addition to these systems'ilities, another type of NFR can substantially impact system design. There are 'design constraints,' which limit freedom of choice for some design options. Examples of design constraints include:

- The system design shall use hardware components only from approved vendors (cyber physical system)
- Signal sign-on shall use the SAML protocol
- Open-source components must be approved in advance by legal
- All user data shall be encrypted and stored in the corporate database
- Java, Python, and Javascript programming languages are approved for general use.
- Any other development languages must be approved in advance.

Of course, in support of innovation, these should be as few as possible and reflect centralized decisions that provide economies of scale, security, or other critical aspects of the solutions set.

The functional requirements, NFRs, and design constraints define the system's scope and quality. By understanding both NFRs and design constraints, teams can make more informed decisions about the system's design, ensuring that it meets the needs of the stakeholders while adhering to any limitations.

Specifying NFRs

[Solution Intent](#) includes NFRs and functional requirements and plays a crucial role in understanding the economics of fixed versus variable solution intent.

Solution intent can also provide traceability links between NFRs, other work items they impact, and tests to verify them. NFRs play a crucial role in understanding the economics of fixed versus variable solution intent (Figure 4).

Figure 4. NFRs are captured in solution intent

Similar to other requirements, some NFRs are fixed and well-known in advance (ex: the adventure ride holds twelve people); others are variable (acceleration at maximum vehicle load shall be no less than x Gs) and will be refined over time.

Like all other requirements, NFRs must be quantified for clarity to ensure everyone clearly understands the goal. Figure 5 provides an example of defining an NFR using some of the properties discussed in [2].

Step 1 defines the NFR's quality, including its name, scale, and method of measure.

Step 2 quantifies the NFR's measurable values, including the current measured value (baseline), the value to achieve (target), and the value that becomes unacceptable (constraint)

Figure 5 includes an example of specifying an NFR for an autonomous vehicle's speed limit detection efficiency. On average, users currently set the speed manually at .1 times per mile, overriding the automated solution. New system functionality should improve that to .01 times per mile, but during implementation should never fall below .15 times per mile.

Figure 5. Steps and an example for specifying NFRs

Figure 5. Steps and an example for specifying NFRs

The following criteria help define NFRs:

Bounded NFRs must have a specific bounded context. For example, an airplane's flight control reliability should be much higher than the infotainment system.

Independent ☐ NFRs should be independent of each other so that they can be evaluated and tested without consideration of other system qualities.

Negotiable ☐ Negotiability of NFRs is a crucial aspect of economic performance.

Testable ☐ NFRs must be testable with objective measures.

NFRs Impact Solution Development

Nonfunctional requirements can have a substantial impact on solution development and testing. As such, Architects and developers should use caution when specifying them. For example, a statement like “99.999 percent availability” may sound great but increase development effort by 10X more than “99.98 percent availability.” The impact of the NFR must be well understood by those defining requirements.

In many cases, applying a [Set-Based Design](#) can [keep options open](#) by specifying NFRs as a range (for example, 99.98 .. 99.999). Teams explore the solution space and gain additional knowledge that leads to better economic decisions. There may indeed be value in 99.999 reliability. Or lower reliability may be more cost-effective with adjustments made elsewhere in the system’s operational environment.

Physical constraints, such as weight, volume, or voltage, impact solution development similarly. Deferring decisions to understand costs and value better often leads to improved economics.

The solution’s [Economic Framework](#) (Figure 6) helps provide criteria to evaluate NFRs. NFRs should consider trade-offs with costs and other considerations. NFRs also affect [Suppliers](#), and their knowledge and concerns should inform NFR specifications and the economic framework.

Figure 6. Five variables guide economic trade-off decisions for NFRs

NFRs may require additional work, either now or in the future. Sometimes they must be implemented all at once. In other cases, teams can take an incremental approach.

All at once ☰ Some NFRs appear as new concerns and will require immediate implementation. For example, regulatory changes may require the organization to respond within the specified time constraints or risk being in violation.

Incremental story-by-story path ☰ At other times, the teams have options. For example, Agile Teams can improve performance incrementally, one story at a time.

Implementation should occur in a way that will allow several learning cycles to determine the right level of NFRs. The structure of an ART can also impact the implementation of NFRs. For example, ARTs organized around architectural layers will find it challenging to implement and test NFRs. Trains with mostly stream-aligned teams designed will find it easier to implement, test, and maintain NFRs. Applying [Agile Architecture](#) practices supports the development of NFRs and helps maintain flexibility as the requirements evolve.

Testing Nonfunctional Requirements

XP advocate and Agile Manifesto [3] co-author Brian Marick helped pioneer [Agile Testing](#) and a testing matrix, which provides a taxonomy for organizing tests. This approach was further developed in Agile Testing [4] and extended to scaling in Agile Software Requirements [4, 5]. Figure 7 describes the latest matrix [6] with

guidance on what to test and when.

Quadrant 4 of the *Agile Testing Matrix* in Figure 7 outlines system quality tests to verify the system meets its NFRs. NFRs often require a suite of specialized automated testing tools (for example, load and performance testing tools) or internal solution telemetry to validate compliance.

Figure 7. NFRs are in quadrant 4 of the Agile testing matrix

Figure 7. NFRs are in quadrant 4 of the Agile testing matrix

Due to their considerable scope and tooling requirements, testing NFRs may require the help of the [System Team](#). Since any system changes can violate conformance with NFRs, these tests must be run continuously, or at least whenever it's practical. To support this built-in quality, teams should automate testing NFRs to run continuously with other tests or on demand when needed.

Learn More

- [1] Nonfunctional requirement. https://en.wikipedia.org/wiki/Non-functional_requirement
- [2] Gilb, Tom. *Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*. 2005.
- [3] Manifesto for Agile Software Development. <http://AgileManifesto.org/>
- [4] Crispin, Lisa, and Janet Gregory. *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison-Wesley, 2009.
- [5] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.
- [6] Gregory, Janet, and Lisa Crispin. *More Agile Testing: Learning Journeys for the Whole Team*. Addison-Wesley, 2015.
- [7] Leffingwell, Dean, and Ryan Shriver. *Nonfunctional Requirements (System Qualities) Agile Style*. Agile 2010.

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite

SAFe videos and presentations

Blog

TRAINING

Course Calendar

About Certification

Become a Trainer

GET SOCIAL

Twitter

Linkedin

YouTube

Permissions Form

Usage and Permissions

300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

Privacy Policy

Cookie Policy

Your California

Consumer Rights

English

Scaled Agile Framework



Agile software development and traditional cost accounting don't match.

ÑRami Sirkia and
Maarit Laanti [1]

Lean Budgets

Lean Budgets is a financial governance approach that funds value streams instead of projects, accelerating value delivery and reducing the overhead and costs associated with traditional project cost accounting.

When implementing Scaled Agile, many organizations quickly realize that the drive for Business Agility through Lean-Agile development conflicts with traditional budgeting and project cost accounting methods. As a result, moving to Lean-Agile development—and realizing the potential business benefits—is compromised, or worse, blocked entirely. To address this problem, SAFe introduces Lean Budgets as its approach to financial governance.

Details

Each SAFe portfolio operates within an approved budget, a fundamental principle of financial governance for developing and deploying Business *Solutions*. Figure 1 illustrates a typical *Enterprise* strategic planning process that creates each portfolio's budget in SAFe.

Figure 1. Portfolio budgeting overview

SAFe's Lean approach to budgeting is significantly different than traditional methods. It gives effective financial control over all investments, with far less overhead and friction, and supports a much higher throughput of development work. Figure 2 illustrates the transition and highlights the three primary steps for adopting Lean budgets.

Figure 2. Moving from traditional to Lean Budgets

Figure 2. Moving from traditional to Lean budgets

The Problem of Traditional Project Cost Accounting

Understanding the problems caused by conventional project cost accounting is critical before describing the adoption of Lean budgets.

Project Budgeting Creates Multiple Challenges

Figure 3 represents the traditional budgeting process for most enterprises. In this example, the enterprise has four different cost centers. Each cost center must contribute some of its budget or people (the primary cost element) to a project.

Figure 3. Traditional project-based cost budgeting and accounting model

Figure 3. Traditional project-based cost budgeting and accounting model

This standard budgeting model creates several problems:

Slow, complicated budgeting process ☰ Many large technology projects are subject to siloed organizational structures and require multiple cost centers and functional managers to fund a single project.

Less accuracy in decision-making ☰ Individuals and teams make poorer choices when forced to make specific decisions too early in the Œcone of uncertainty, when the least amount of learning has occurred. There is no time or budget to identify and validate assumptions and create experiments that provide the data needed to determine if and how the portfolio should implement a solution.

Temporary teams lower overall performance ☰ People are generally assigned to a project temporarily and then return to their functional silo for future assignments to a new initiative. This structure hinders learning, employee engagement, and overall organizational performance.

Waiting on specialists causes delays to value delivery ☰ Traditional project teams focus on individual skills, and it's common for one project to block another while waiting for the availability of specialists. If a task takes longer than planned—

which it often doesÑmany people will have moved on to other projects, causing further delays and lower quality.

Full resource utilization is favored over a fast flow of value Ð In the pursuit of efficiency, everyone is assigned to 100% capacity, often to multiple projects. However, Reinertsen notes that Òoperating a product development process near full utilization is an economic disasterÓ [2]. This disaster results from long queues, project delays, and high variability between forecasted and actual time and costs. A study by Adler concludes that Òif managers had reduced their planned utilization rate to 80%, they could have reduced development times by 30% or more.Ó [3].

The Project Funding Model Impedes Adaptability

Once the project is underway, the challenges continue as the business needs to change and the resulting project change. However, because the budget and people are fixed for the projectÓs duration, an organization cannot change the plan without the overhead of re-budgeting and reallocating personnel (Figure 4).

Figure 4. Project funding inhibits the ability to adapt to change

Figure 4. Project funding inhibits the ability to adapt to change

Project Delays Happen. Things Get Even Worse.

Often, work will take longer than planned because of new learning, insights, and opportunities. Further, even when things go well, stakeholders may want more of a specific feature. Many organizations manage change through a change control board, adding even more delays and decision-making overhead. The project model also hinders cultural change, transparency, and solution development progress (Figure 5).

Figure 5. When overruns happen, project accounting and re-budgeting increase the cost of delay

When a schedule overruns for any reason, it's necessary to analyze the variances, re-plan, and adjust the budget after getting approval(s) to continue. People are scrambled to different projects, adversely impacting other projects. Now, the blame game starts, causing project managers to fight against each other and financial management against the teams. The ultimate result is information hiding, loss of productivity, and lower employee morale.

Projects Can Stifle Innovation

Solution development also requires innovation, and we cannot innovate without taking risks [2]. Because innovation contains a higher degree of uncertainty, estimating these projects is, at best, challenging. At worst, organizations tend to minimize investments in innovation, eroding the value of the solutions they create. In addition, it can be culturally challenging to stop a project that doesn't realize its stated objectives (a "failed project"). In contrast, the ongoing investment in

innovative solutions promotes a growth mindset where the fast failure of experiments is considered integral to learning.

Beyond Project Cost Accounting with SAFe

SAFe provides a Lean budget approach, which reduces the overhead and costs associated with traditional cost accounting and empowers people through [Principle #9, Decentralized decision-making](#). With this new way of working, the portfolio no longer plans work for others, nor do they track the cost of the work by discrete projects. There are three main steps to Lean budgets, as described below.

1. Funding Value Streams, Not Projects

The portfolio budget funds a set of [Development Value Streams](#). Each delivers one or more business solutions and is given a budget for its value stream (Figure 6). Ideally, [Participatory Budgeting](#) is adopted, engaging a broader stakeholder group. Lean Budget Guardrails define a portfolio's spending policies, guidelines, and practices. Like any good governance, guardrails enable increasing autonomy of teams.

Figure 6. Each value stream has an operating budget for people and other resources

Funding value streams vs. projects delivers several benefits:

- Empowers local content authority** ⚡ moves decisions to where the information lives, enabling faster and better decision-making
- Improves transparency** ⚡ offers clarity of spending through value stream budgets
- Better visibility** ⚡ makes the flow of portfolio business epics and enablers visible from idea through implementation using the portfolio Kanban.
- Improves productivity of knowledge workers** ⚡ working in long-lived, stable value streams and ARTs are more productive than temporary project teams
- Self-organization** ⚡ allows moving people to the most critical work without escalation to management.
- Better management of budgets** ⚡ gives more autonomy to [Agile Release Trains \(ARTs\)](#) and [Solution Trains](#) while providing the proper oversight and flexibility.

In most cases, the expenses for a [PI](#) are fixed or easy to forecast (Figure 7). Moreover, features that take longer than expected *do not* change the budget. As a result, all stakeholders know the anticipated spending for the upcoming period,

regardless of the features implemented.

Figure 7. The budget for a PI is fixed

Figure 7. The budget for a PI is fixed

2. Guiding Investments by Horizon

As a personal financial portfolio balances investments in different asset classes (stocks, bonds, real estate), SAFe balances solutions in different investment horizons. Get this balance wrong, and you can starve the future by over-investing in today or miss near-term opportunities while allocating too much money into an uncertain future.

Adapted partly from the McKinsey model [4], the SAFe solution investment horizon model highlights spending allocations for value stream solutions (Figure 8). This model helps value stream owners and others accountable for financials to make more informed investment decisions and helps align the portfolio with strategic themes while promoting overall health and growth.

Figure 8. SAFe investment horizon model illustrating solution investments by horizon

The following paragraphs provide an overview of each horizon.

Horizon 3 (Evaluating): Horizon 3 investments are dedicated to investigating new potential opportunities for profitable growth in the future, *typically within 3-5 years*. These could be innovative new solutions and other investments that could even represent changes to the fundamental business model. Generally, these exploratory and research activities require modest funding and can often be somewhat isolated from the current operating model. Accordingly, an epic is usually created to start the initiative. If the epic hypothesis is proven true and the emerging solution provides a sufficiently compelling return on investment, it will typically continue to horizon two.

Horizon 2 (Emerging): Horizon 2 reflects the investments in promising new solutions from horizon three. These investments are anticipated to provide a profitable return *within 1-2 years*. Since these new solutions are promising, the business is willing to make ongoing investments above the current return. Some may require horizon one resources, and the portfolio must ensure they

have sufficient operating resources to reach horizon one. Suppose the decision is made to stop, likely. In that case, some modest investment is still necessary to *decommission* the solution, as the horizon two solutions have usually made their way into the internal and external business ecosystem.

Horizon 1: Horizon one reflects the desired state where solutions deliver more value than the cost of the current investment. These solutions require ongoing investment to maintain and extend functionality. For convenience in reasoning about these investments, horizon one is divided into two profiles:

Investing: These investments reflect solutions that require significant ongoing investment due to market or solution immaturity, changes to the market or technology, or the desire to fuel growth, such as capturing market share in a fast-growing product.

Extracting: These investments typically represent stable solutions delivering great value with a lower need for additional spending. Investment in these solutions ensures enduring value, profit, and cash flow, enabling the funding of emerging solutions.

Horizon 0 (Retiring): All solutions eventually meet end-of-life. Horizon 0 reflects the investment needed to decommission a deployed solution, which frees the budget for more promising investments in other horizons.

Value stream leaders must learn to manage all four horizons simultaneously. After all, value streams must dynamically evolve solutions, introduce and retire solutions, manage technological change, and respond to market demands. In addition, the portfolio occasionally must create an entirely new value stream, and others may be retired. The portfolio budget guardrails constrain these investment choices.

3. Applying Participatory Budgeting

SAFe [Participatory Budgeting](#) (PB) is an LPM event in which stakeholders decide how to invest the portfolio's budget across solutions and epics, as illustrated in Figure 9. PB ensures that value streams receive the funding required to advance the solutions and promotes the collaborations that help right-size the investments and align strategy and execution.

PB provides numerous benefits:

- Provides leaders with insights and perspectives from multiple stakeholders
- Creates alignment on difficult funding choices
- Improves engagement and morale
- Reduces implementation time and overhead

Stakeholders are organized into groups of four to eight people during a PB event. Each group should have a mix of roles from different value streams to promote understanding. Each participant gets the list of solutions and epics, the amount of investment funding requested by the value streams, and an equal portion of the total portfolio budget. For example, consider a portfolio with total requested investment funding of 46M and an allocation of 40M. A forum of five participants would allocate 8M to each participant.

Participants collaboratively invest their budgets against the requests under the guidance that solutions and epics should generally be fully funded to be considered for actual funding when the forum is complete. Since the participants can't support all the items, they must work together to identify the best investments. Even more importantly, participants from different value streams must collaboratively pool their budgets to support initiatives that no single value stream can fund. Because partially funded solutions and epics are candidates for termination, the group will negotiate to determine where to make the best investments. The discussions from this collaboration allow participants to make choices that optimize value delivery across the portfolio.

Figure 9. Participatory budgeting overview

During or after PB forums (sessions), teams discuss how their choices adhere to investment horizon guardrails and adjust their investments accordingly. These sessions also enable teams to make recommendations that can increase or decrease the investment in a solution or epic that reflects the wisdom and experience of the entire group. The results of multiple forums are analyzed, and LPM finalizes any adjustments needed to the value stream budgets in alignment with agreed-upon funding.

Learn More

[1] Special thanks to Rami Sirkia and Maarit Laanti for an original white paper on this topic, which you can find [here](#).

[2] Reinertsen, Don. *Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

[3] Getting the Most Out of Your Product Development Process.

<https://hbr.org/1996/03/getting-the-most-out-of-your-product-development-process>

[4] Baghai, Mehrdad, and Steve Coley. *The Alchemy of Growth: Practical Insights for Building the Enduring Enterprise*. Basic Books, 2000.

Last update: 7 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

[Course Calendar](#)

Weekdays: 9am to 5pm

[About Certification](#)

Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



People at work are thirsting for context, yearning to know that what they do contributes to a larger whole.

~Daniel Pink

Portfolio Vision

The Portfolio Vision describes the future state of a portfolio's value streams and solutions.

[Adjust Cookie Settings](#)

The vision is a critical input to identifying the potential epics that will be needed to achieve the future state.

Details

The portfolio vision sets a longer-term context for near-term decisions in a practical and inspirational way, clearly articulating why the future state of enhanced products and services is worth achieving. Understanding the longer-term view helps [Agile Teams](#), [Agile Release Trains](#), and [Solution Trains](#) make more informed choices about the development of functionality in both the short and long run.

[Lean Portfolio Management \(LPM\)](#) is primarily responsible for ensuring the strategic direction of the portfolio maps to the strategic themes and enterprise strategy. This requires a clear understanding and communication of the portfolio vision. In the book *Switch* [2], authors Dan and Chip Heath liken this future vision to a destination postcard, as Figure 1 illustrates.

Figure 1. A postcard from the future is a way to describe the portfolio vision

Figure 1. A postcard from the future is a way to describe the portfolio vision

The portfolio vision should have the following characteristics:

Aspirational, yet realistic and achievable ☐ It must be compelling and somewhat futuristic, yet practical enough to be feasible over some meaningful timeframe

Motivational to engage others on the journey ☐ The vision must align with the Strategic Themes, as well as with the individual team's purpose

[Business Owners](#) or senior leaders typically present this longer-term view and business context during the [PI \(PI\) Planning](#) event. These leaders can inspire and align the teams, increasing engagement and fostering creativity to achieve the best results.

Introducing the SAFe Portfolio Canvas

The SAFe portfolio canvas (Figure 2) is based on the Business Model Canvas (see this topic in the [Enterprise](#) article) developed by Alexander Osterwalder [1]. The portfolio canvas defines the [Development Value Streams](#) included in a SAFe portfolio, the value propositions and the [Solutions](#) they deliver, the customers they serve, the budgets allocated to each value stream, and other vital activities and events required to achieve the portfolio vision.

Figure 2. The portfolio canvas

Figure 2. The portfolio canvas

[Download](#)

The portfolio canvas and its blocks are described below.

1. Value Propositions

The value propositions describe the customers and the value delivered by the solutions of each value stream, as well as the customer segments and relationships, budget, and KPIs /revenue. Create a separate row for each development value stream.

Value Streams: The development value streams used to build the systems and capabilities that enable business processes in operational value streams or provide products and services to the operational value streams.

Solutions: Each value stream produces one or more solutions, which are the products, services, or systems delivered to the customer, whether internal or

external to the enterprise.

Customers: Customers describe the internal or external customers for each value stream. It defines how the business views and treats various sets of customers differently based on their common attributes.

Channels: Explains how the enterprise delivers its products and services to intermediaries, customers, and end-users. If serving external customers, this may include marketing and sales mechanisms used to reach customers (e.g., web, direct sales, brick-and-mortar store, distribution network). If serving internal customers, it captures the interfaces with internal stakeholders and end-users (for example, internal websites or custom IT applications).

Customer Relationships: The types of customer relationships needed to effectively apply and leverage the business's products and services. It describes the connections and communications with each customer segment. These relationships influence the design of solutions and the allocation of resources within the portfolio.

Budget: Each value stream is assigned a [Lean Budget](#), which includes operating, overhead, and capital expenses.

KPIs / Revenue: Key Performance Indicators (KPIs) define the measures used to evaluate the results of the value stream investment.

2. Resources and Activities

The resources and activities describe the key partners, activities, and other resources needed to achieve the value propositions.

Key Partners ▷ The various buyer-supplier relationships and business alliances that facilitate achieving the value proposition.

Key Activities ▷ The enterprise's most important actions to deliver its products and services.

Key Resources ▷ The enterprise's critical physical, intellectual, financial, human, and other capabilities and assets used to achieve its objectives.

3. Cost Structure and Revenue Streams

The cost structure and revenue streams describe how the portfolio's costs are structured and define how revenue or value is achieved.

Cost Structure: Identifies the most significant costs in the portfolio's business model, including the structural aspects, such as license costs, development labs, and costs of external services. Building cyber-physical systems also have other costs (for example, hardware and firmware), which must be considered.

Revenue Streams: If the development value streams monetize directly, list the types and sources of revenue from customers. Note the major sources of revenue and how the customer is charged (fixed price, usage-based, and so on). For internal customers, non-profits, and government agencies, describe the value of the solutions.

Capturing the Current State of the Portfolio

The current state canvas represents the as-is state for the portfolio, enabling alignment of the organization on its structure, purpose, and status. One way to capture the current state is to assemble one or more teams to create a shared understanding. The team should include the Agile Release Train (ART), value stream [Business Owners](#), Lean Portfolio Management (LPM), [Epic Owners](#), and Architects. RTEs, [Product](#) and [Solution Management](#), [Product Owners](#), and other portfolio stakeholders

A straightforward approach to capturing the current state is to iterate through each canvas's building blocks, summarizing the essential aspects. Teams typically use sticky (virtual or physical) notes with a few keywords to fill out each building block (Figure 3). The real power of the canvas is its ability to represent the entire portfolio on one page and to gain insights.

As part of filling out the current state portfolio canvas, it can be helpful to reason further about each value proposition. One way to do this is to create separate [Value Stream](#) canvases (see this topic in the value stream article), which are then rolled up to the portfolio (Figure 3). The appropriate stakeholders in each value stream should develop their Value Stream Canvas.

Figure 3. Creating the portfolio canvas from a set of value stream canvases

Figure 3. Creating the portfolio canvas from a set of value stream canvases

Envisioning the Future State

The next step is to envision the future state, which helps define the vision for the portfolio. The difference between the current and future state represents the gap, translated into the vision to attain the future state.

Understanding Opportunities and Threats

There are many tools and techniques to help understand the opportunities for the future state. Some of these techniques include the ***SWOT analysis*** and ***TOWS strategic options matrix***, which can help develop a plan for the future so that the business can survive and thrive in the long run.

The SWOT and TOWS analysis are meant to be used together. The SWOT analysis (Figure 4) is used to identify the strengths, weaknesses, opportunities, and threats related to the current business situation.

Figure 4. SWOT analysis

Figure 4. SWOT analysis

The information from the SWOT analysis is used to fill in the strengths, weaknesses, opportunities, and threats of the TOWS strategic options matrix (Figure 5) using the four outer boxes (for example, Internal Strengths, External Opportunities)

The critical difference between the SWOT and TOW analysis is the outcomes that they create. A SWOT analysis is a great way to uncover the current situation of your value stream, product, or portfolio. At the same time, TOWS is used primarily for identifying strategic options to create a better future state.

Figure 5. TOWS Strategic Options Matrix [3]

Figure 5. TOWS Strategic Options Matrix [3]

Evaluating Alternatives to Determine a Future State

The portfolio's [Strategic Themes](#) and SWOT and TOWS analysis are critical inputs to exploring alternatives for the future state. LPM uses the current state portfolio canvas as a starting point to explore how the portfolio could evolve in alignment with the strategic themes. Start by selecting a specific block in the portfolio canvas, identifying a potential change or opportunity, and then explore how it impacts the other parts of the canvas (Figure 6).

Figure 6. Exploring different scenarios to evolve a solution portfolio

Figure 6. Exploring different scenarios to evolve a solution portfolio

For example, consider a portfolio for a printing company that has invested heavily in the on-demand printing of books. The value stream may want to capture a different customer segment as the foundation of the future state (such as customized books for schools) or may explore a change in customer relationships (such as moving to a self-service model). Each of these changes may impact other blocks on the canvas.

Some changes to achieve the future state will require investing in large initiatives. The teams must create business and enabler epics that feed directly into the [Portfolio Backlog](#).

Maintaining the Portfolio Vision

The creation of the vision is not a once-and-done exercise. It is updated by using the current and future state Portfolio Canvas. As new information about the evolving set of portfolio solutions emerges, portfolio stakeholders periodically

review and update the canvas and the vision. Other triggers include the introduction of new solutions, mergers and acquisitions, the market rhythms of the portfolio defined in the portfolio [Roadmap](#), and other strategic changes that may affect the portfolio's value streams or solutions.

Learn More

[1] Osterwalder, Alexander, and Yves Pigneur. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, 2010.

[2] Heath, Chip, and Dan Heath. *Switch: How to Change Things When Change Is Hard*. Broadway Books, 2010.

[3] <https://brungerblog.wordpress.com/2016/03/20/tows-matrix-for-marketing-brainstorming/>

Last update: 17 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Linkedin](#)

[YouTube](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*Everything should
be made as simple
as possible, but not
simpler.*

ÑAlbert Einstein

Enterprise Architect

The Enterprise Architect is responsible for establishing the portfolio's technology vision, strategy, and roadmap.

Enterprise Architects provide the vision, evolution, and communication of an enterprise's technical architecture, creating the strategy and portfolio-level

technical roadmaps for new and innovative technologies. This often includes incorporating [Cloud](#), [Big Data](#), and [AI](#) technologies to create strategic advantage.

EAs collaborate with [System Architects](#) and [Solution Architects](#) to guide the technical design of [Solutions](#) developed by [Agile Release Trains \(ARTs\)](#) and [Solution Trains](#). They lead [Enabler Epics](#) through the Portfolio Kanban, which builds the more significant architecture for the portfolio's solutions. Relying on continuous feedback, EAs foster adaptive design and engineering practices and drive ARTs and teams to unite around a shared technical vision.

Details

EAs guide the portfolio's value streams as they build new elements of an organization's enterprise architecture. For example, its data and information, applications, technologies, and finding ways to use these architectural elements to meet the company's organizational standards and improve its overall performance.

Good strategic technical planning, communication, and visibility can result in optimal system performance, significantly improving [Business Agility](#). EAs provide some [Architectural Runway](#) and governance, supporting current and future needs, such as usability and behavioral constructs across the enterprise's solutions. System and Solution Architects offer this guidance for Agile Release Trains (ARTs) and Solution Trains.

Collaboration Among Architecture Roles

SAFe defines three architect roles, Enterprise, Solution, and System Architect, that address the [Portfolio](#), [Large Solution](#), and [Essential](#) levels. These architects regularly collaborate to ensure alignment and address issues and concerns as they arise. Figure 1 illustrates a high-level overview of their responsibilities.

Figure 1. Comparison of SAFe architecture roles.

Figure 1. Comparison of SAFe architecture roles.

In addition, the relationship between business and technology strategy requires active collaboration between architects and other SAFe roles to ensure that the architecture meets the current and evolving needs of the business and its customers.

Enterprise Architecture Strategy

The strategy for enterprise architecture enables embracing organizational change faster, providing a significant competitive advantage (Figure 2).

Figure 2. Five elements of enterprise architecture strategy

Figure 2. Five elements of enterprise architecture strategy

Choice of technology and usage ☰ Choosing appropriate technologies is a critical element of strategy development. Supporting activities include research and prototyping, understanding applicability and scope, and assessing the maturity of innovative new technologies.

Solution architecture strategy ☰ The Enterprise Architect works closely with the Solution and System Architects to ensure that individual program and product strategies align with business and technical objectives. For example, emerging solutions to local problems should be consistent with the overall enterprise strategy. When that's not the case, decisions should be explicit, as the inconsistent option may well influence future enterprise strategy.

Infrastructure strategy ☰ Developing and maintaining plans for infrastructure can be more challenging, as it overlaps with the work of System Architects. Some of these responsibilities include the reuse of configuration patterns, common physical infrastructure, knowledge sharing across ARTs and Solution Trains, and especially [System Teams](#). Also, some of the development and deployment infrastructure will likely intersect with internal IT systems. The Enterprise Architect can provide direction there as well.

Inter-program collaboration ☰ Standard design and infrastructure practices help simplify and align architecture across different teams and programs. However, it's also essential that value streams and ARTs have sufficient degrees of freedom. Otherwise, innovation decreases. Therefore, both standard and variable architectural designs should be actively discussed and shared among the ARTs.

Implementation strategy ☰ The importance of an active, incremental Agile implementation strategy can hardly be overstated. Building the technical foundation for business epics into the architectural runway must be a gradual process. Continuous learning and fast feedback allow architecture and business functionality to grow synchronously over time. This rapid pace of change requires [Agile Teams](#) to refactor their code as necessary and to preserve multiple possible design options wherever practical. Abstraction and generalization help avoid binding specificity too early, which maintains architectural flexibility for future business needs.

Responsibilities

The primary responsibilities of Enterprise Architects fall into four main areas, as illustrated in Figure 3. Each responsibility area is described in the sections that follow.

Figure 2. Enterprise Architects' areas of responsibility

Figure 3. Enterprise Architects' areas of responsibility

Aligning Business and Technical Strategies

Enterprise Architects are critical for understanding the connection between new technology trends and business. They help the organization identify, evaluate, and respond to opportunities and trends in its broader ecosystem. An enterprise's complexity needs to be governed to make innovation easier. Otherwise, change becomes nearly impossible if the architecture is tightly coupled. EAs can create room for innovation by reducing this complexity and facilitating change. For example, they are actively involved in the following:

Aligning organization design with optimal architectural design ⚡ The organizational design of [Agile Teams](#) and Agile Release Trains (ARTs) should align with the desired technical architecture, not the other way around.

Defining Strategic Themes ⚡ Since EAs have a broad knowledge of technologies, business domains, reference architectures, security, design thinking, architectural frameworks, governance, risk, and compliance, their input is vital to creating [Strategic Themes](#). EAs collaborate with [Lean Portfolio Management \(LPM\)](#), enterprise

executives, and portfolio stakeholders to develop the strategic themes for a specific portfolio, as illustrated in Figure 4.

Figure 4. EAs collaborate with enterprise and portfolio stakeholders to create strategic themes

Communicating vision and business Strategy △ EAs need to understand and communicate Strategic Themes and other business drivers for architecture to System and Solution and non-technical stakeholders.

Responding to emerging opportunities and threats △ Providing strategic technical direction across value streams, EAs ensure the organization can take advantage of emerging opportunities while responding to and mitigating threats.

Offering technical recommendations △ EAs provide guidance and suggestions for developing and delivering technology stacks, managing interoperability and application program interfaces (APIs), hosting and cloud computing strategy, and governing artificial intelligence (AI) and big data.

Maintaining relationships △ EAs should create and maintain personal connections with ART and Solution Train Architects and Agile Teams through the following types of tasks and events:

- Getting feedback on current enterprise-wide initiatives
- Participating in architecture and design [Communities of Practice](#)
- Attending [System Demos](#) whenever critical redesign or architectural work is in progress
- Participating in PI Planning events to guide the implementation of enabler epics and features.

Identifying value streams Ⓛ EAs assist the Value Management Office (VMO) and the [Lean-Agile Center of Excellence \(LACE\)](#) in identifying and designing [Development Value Streams](#) that optimize the flow of value for new and existing technology initiatives.

Enabling Built-in Quality Ⓛ The Systems Team helps realize the architectural vision by building the supporting infrastructure that enables Agile Teams to design, implement, test, and deliver value using [Built-In Quality](#). EAs coordinate with System and Solution Architects to ensure their solutions align with the broader technical vision.

Establish the Portfolio's Intentional Architecture

EAs help organizations respond to new business challenges with larger-scale architectural initiatives that require some intentionality and planning. This intentionality provides Agile Teams with a solid foundation to work independently, allowing the proper balance of intentional architecture and emergent design. This work typically involves:

Maintaining the architectural Runway Ⓛ EAs lead the strategy for building and maintaining the architectural runway. One way they do this is by serving as Epic Owners for portfolio Enabler Epics. They also closely collaborate with System and Solution Architects and Agile Teams to guide the evolution of solutions, existing code, and technical infrastructure needed to implement near-term features without excessive redesign and delay.

Aligning technology approaches across ARTs Ⓛ EAs actively engage with System and Solution Architects, ensuring emergent design choices are made with an understanding of the overall architectural strategy and roadmap, minimizing technology complexity, and avoiding unnecessary duplication of capabilities or solutions.

Setting technology direction Ⓛ EAs work with LPM and the Value Management Office (VMO) to determine technology direction. EAs define and communicate the architectural [Roadmap](#), ensuring the business invests in the right systems and services to support current and future business needs.

Coding and pattern Reuse Ⓛ EAs work with System and Solution Architects and [Agile Teams](#) to help reuse code and existing design patterns to implement new business and technology functionality. They also influence modeling, design, and coding practices.

Synchronizing technical disciplines Ⓛ During a PI, EAs help synchronize various

disciplines across solutions whenever applicable: system and data security and quality, production infrastructure, solution user experience ([Lean UX](#)), and [Nonfunctional Requirements \(NFRs\)](#).

Defining organizational structures that promote the desired architecture ⓘ

Understanding Conway's Law is essential to organizational design. According to Conway, organizations design systems that closely match the organizational structures in the hierarchy (see [Business Agility](#)). Instead, use the system's future-state design to influence the organizational design, specifically when identifying value streams (see [SAFe Principle #10, Organize Around Value](#)) and optimizing teams on the ART with [Team Topologies](#). This evolution is known as the Inverse Conway Maneuver.

Figure 4. Define organizational structures that promote the desired architecture

Figure 5. Define organizational structures that promote the desired architecture

Rationalizing the Technology Portfolio

EAs help rationalize the technology portfolio by reviewing and streamlining the existing application portfolio to improve efficiency, reduce complexity, and lower the total cost of ownership (TCO).

Application rationalization improves the overall effectiveness of IT by simplifying the technical landscape. This helps ensure that organizations are not running redundant applications, overspending on license costs, hardware, networking, or unnecessary infrastructure. EAs typically work with System and Solution Architects to:

Eliminate redundancy D Consolidating similar applications reduces IT spending.

Minimize unnecessary IT spending D With a strong application rationalization strategy and continual maintenance, a well-organized technical architecture will provide the data needed to help define the necessity of future applications.

Reduce IT cost D It's not uncommon to uncover substantial savings after application rationalization. This saving can be used to reinvest in innovative endeavors such as artificial intelligence and machine learning, big data, and cloud initiatives.

Reduce complexity D Eliminating unused applications and moving to the cloud can reduce complexity via centralized management, automation, hardened security, consistent tools, technologies, and processes.

Support procurement D EAs can help streamline procurement and renewal planning as organizations review existing purchasing structures, products, license agreements, and other transactions.

Reduce training overhead D Every application used at an organization requires vendors or in-house staff support, which can be expensive. If there are fewer applications to support, there will be less money spent on maintaining these apps.

Fostering Innovative Ideas and Technologies

Decentralized decision-making and execution are hallmarks of SAFe. To enable the decentralization of architecture while maintaining Solution integrity, EAs support the establishment and evolution of architectural standards. These standards help provide alignment on how to build solutions the right way. EAs, in close partnership with System and Solution Architects, establish, foster, and evolve architectural standards and apply them in the following ways:

Achieving business outcomes D Ensure that the portfolio can effectively reach its desired business outcomes by using new technologies, practices, and standards.

Aligning technical decisions D It can be challenging to align decision-making for technology across one or more portfolios. EAs help coordinate and develop common standards and architectures for system and data security, quality, production infrastructure, and Nonfunctional Requirements (NFRs). These include internal enterprise architecture standards and external ones such as TOGAF, DODAF, and others where applicable.

Supporting the development environment D Facilitating the reuse of code, components, and proven design patterns. Influencing and promoting good practices for modeling, system design, and coding. Supporting the [Continuous Delivery](#)

[Pipeline](#), [DevOps](#), and other flow capabilities (see [Portfolio Flow](#)) to get faster customer feedback and time to market.

Retiring obsolete technologies ☰ Helps the portfolio’s value streams migrate off outdated technologies and decommission solutions that are no longer fit for purpose.

Guiding Enabler Epics

Enabler epics are often created by EAs who support the portfolio backlog and steer them through the Portfolio Kanban system, guiding their analysis and the information needed to estimate and implement them. For example, their role includes:

Acting as an Epic Owner ☰ EAs serve as [Epic Owners](#) for architectural initiatives, leading enabler epics through the Portfolio Kanban and their subsequent implementation.

Splitting epics into features ☰ Collaborate with [Product](#) and [Solution Management](#) and System and Solution Architects to decompose enabler epics into enabler features and capabilities and help prioritize these backlog items in their respective ART and Solution Train backlogs.

Engaging in SAFe events ☰ Participate in [PI Planning](#), System, and [Solution Demos](#) whenever a critical activity is related to an enabler epic.

Enterprise Architects as Lean-Agile Leaders

The SAFe [Lean-Agile Mindset](#) promotes a healthy work environment where everyone operates on facts, verifies assumptions, and reduces uncertainty through frequent experiments. This mindset is critical for EAs, who work one (or two) steps removed from day-to-day development activities. They have the most significant impact by teaching, mentoring, and helping improve the effectiveness of Agile Teams rather than directly specifying the solution designs. Instead, they contribute to the technical [Vision](#) and roadmap to chart a course for the solution.

Now more than ever, Business Agility is needed to compete and thrive in the *age of digital and software*. This new reality requires an adaptive technology strategy, which in turn requires a Lean-Agile approach to enterprise architecture.

EAs who can adapt to new ways of thinking and working and stay up-to-date on the latest technological developments will be essential to the organization’s success. They also act on the human system that creates the technology to ensure greater agility and effectiveness. They are Lean-Agile Leaders responsible for mentoring teams and enhancing the overall capabilities of contributors. This new generation of EAs will need to lead by example, continuously learn, and model SAFe’s Lean-Agile mindset, values, principles, and practices. They help lead the change to the

new way of working and organizing Agile Teams and ARTs around value.

EAs ensure the organization operates effectively by participating as a liaison or member of the Lean-Agile Center of Excellence (LACE). They contribute to Value Stream Identification and Mapping workshops, coach engineers in [Technical Agility](#), and actively participate in the adoption of SAFe.

Learn More

[1] Bloomberg, Jason. *The Agile Architecture Revolution*. Wiley, 2013.

[2] Coplien, James, and Gertrud Bjørnvig. *Lean Architecture: for Agile Software Development*. Wiley, 2010.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS

[YouTube](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Course Calendar](#)

[Privacy Policy](#)

[About Certification](#)

[Cookie Policy](#)

[Become a Trainer](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The big question of our time is not can it be built? But should it be built? This places us in an unusual historical moment: our future prosperity depends on the quality of our collective imaginations.

—Eric Ries

[Adjust Cookie Settings](#)

Epic Owners

The Epic Owner is responsible for coordinating epics through the portfolio Kanban system.

Epic Owners (EO) work collaboratively with stakeholders to define the [Epic](#), the Lean business case, and the definition of a Minimum Viable Product (MVP) and is responsible for shepherding the epic through the portfolio kanban system. If [Lean Portfolio Management](#) (LPM) approves the epic, the EO coordinates the introduction of the epic into implementation with the [Agile Release Trains](#).

Details

Before being committed to implementation, epics require analysis, a Lean business case, and a definition of the MVP. Epic Owners take responsibility for the crucial collaborations needed to guide epics through this process. [Enterprise Architects](#) typically act as Epic Owners for [Enabler](#) epics. Epics often require the collaboration of multiple Agile Release Trains (ARTs) to implement them. The Epic Owner coordinates this work.

Collaboration is a Key Aspect of the Role

Epic Owners can come from anywhere in the enterprise. But they can only be effective by collaborating closely with other stakeholders. They help fill in the gaps in planning and execution that otherwise occur. Figure 1 highlights the people who typically fill the Epic Owner role and the key collaborations they foster.

Figure 1. Epic Owner collaboration with key stakeholders

Figure 1. Epic Owner collaboration with key stakeholders

Typically, an Epic Owner works on epics within their area of expertise and current business mission. Epic Owners are responsible for formulating and elaborating the epic and analyzing its cost and impact by collaborating closely with these groups.

Responsibilities

Figure 2 illustrates the Epic Owner's four primary areas of responsibility.

Figure 2. The four primary responsibilities of the Epic Owner

Figure 2. The Epic Owner's four areas of responsibility

The following sections describe each of these responsibilities.

Guiding Portfolio Epics

The Epic Owner guides epics through the [Portfolio Kanban](#) system from identification through implementation, collaborating closely with stakeholders and subject matter experts.

Figure 3 illustrates the role of the Epic Owner in each state of the Portfolio Kanban system.

Figure 3. The Epic Owner's role as they guide the epic through the portfolio Kanban
Creating the Lean Business Case

Epics that make it to the *analyzing* state require more rigorous exploration, a Lean business case, and further investment to evaluate their cost and benefits. This analysis typically requires active collaboration among the following roles:

- Business Owners
- System Architects and Solution Architects
- Product and Solution Management
- Agile Teams

Epic Owners, stakeholders, and internal teams collaborate to size the epic and provide input for economic prioritization based on [Weighted Shortest Job First \(WSJF\)](#), the Lean business case, and other relevant information.

Epic Owners take responsibility for the critical collaborations needed to analyze the epic and create the Lean business case. At the same time, Enterprise Architects

typically coordinate the enabler epics that support the technical considerations for business epics.

The Epic Owner is primarily responsible for creating and presenting the Lean business case to Lean Portfolio Management (LPM) for a ÔgoÕ or Ôno-goÕ decision. Approval, however, is not guaranteed, as enterprises typically have more ideas and opportunities than they can execute. Accordingly, when presenting the epic, the Epic Owner should focus on the merits of the business case, secure in the knowledge that the collaborative discussions that form the foundation of LPM will ensure they are making optimal investment choices.

Supporting the MVPÕs Development

After the epic is approved, the Epic Owner works with ARTs to initiate the MVPÕs development activities, following the SAFe Lean Startup Strategy to evaluate the business outcome hypothesis.

This strategy recommends a highly iterative build-measure-learn cycle for product innovation and strategic investments. It provides the economic and strategic advantages of a Lean startup (see [Epic](#)) by managing investment and risk incrementally while leveraging the flow and visibility benefits of SAFe.

Developing the MVP and gathering the data necessary to prove or disprove the epic hypothesis is a highly iterative process, continuing until positive results are obtained, or the teams consume the investment allocated for the MVP. The output of a proven hypothesis is an MVP suitable for getting customer feedback that would warrant further investment by the value stream. If disproven, investment in the epic stops.

Coordinating the EpicÕs Development

If the epic hypothesis is proven, Epic Owners often play a significant role in coordinating its implementation across value streams where they:

- Collaborate with Product and Solution Management and System and Solution Architects to decompose the epic into features and capabilities and help prioritize these backlog items in their respective ART and Solution Train Backlogs
- Participate in PI Planning, [System Demo](#), and [Solution Demo](#) whenever there is critical activity related to the epic
- Work with Agile Teams that perform research spikes, create proofs of concept, mock-ups, etc.
- Coordinate and synchronize epic-related activities with sales, marketing, and other business units
- Facilitate the implementation of the epic through the continuous delivery pipeline

and release on demand

Understand and report on the epic's progress to key stakeholders and LPM

The role of the Epic Owner typically continues until the ARTs have sufficiently integrated the epic into their roadmaps and the Epic Owner's expertise or coordination is no longer required.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 22 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

TRAINING

[Course Calendar](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

[BUSINESS HOURS](#)

[About Certification](#)

Weekdays: 9am to 5pm

[Become a Trainer](#)

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Sound strategy starts with having the right goal. Strategy is about making choices and trade-offs; it's about deliberately choosing to be different.

—Michael Porter,
author and Harvard
University Professor

Adjust Cookie Settings

Strategic Themes

Strategic themes are portfolio-level business objectives that provide competitive differentiation and strategic advantage. They provide business context for portfolio strategy and decision-making, representing aspects of the enterprise's strategic intent.

Details

Enterprise executives and [Lean Portfolio Management \(LPM\)](#) work together to analyze various inputs to establish a set of strategic themes (Figure 1) for each SAFe [Portfolio](#). These *differentiated* business objectives communicate aspects of strategic intent from the enterprise to the portfolio.

Figure 1. Enterprise and portfolio stakeholders collaborate to create strategic themes

Strategic themes align the business strategy of the Enterprise (or [Government](#) agency) to a SAFe [Portfolio](#), as Figure 2 illustrates. Since they reflect enterprise strategy, they tend to be somewhat stable for a period. Annual or twice-annual updates seem to work for most enterprises.

Figure 2. Strategic themes connect the enterprise strategy to a portfolio

Figure 2. Strategic themes connect the enterprise strategy to a portfolio

The Influence of Strategic Themes

Since strategic themes influence portfolio strategy and provide business context for portfolio decision-making, they affect many aspects of the SAFe implementation, as illustrated in Figure 3.

Figure 3. The influence of strategic themes across a portfolio

Figure 3. The influence of strategic themes across a portfolio

The following sections describe how strategic themes influence each aspect shown in Figure 3.

Portfolio Vision and Backlog

Strategic themes are direct inputs to the portfolio vision. They may influence the solutions, partners, key activities, customer segments, revenue streams, and other business model elements in the portfolio canvas.

They also provide insight into the [Epics](#) necessary to achieve the vision, serving as decision-making criteria in the [Portfolio Kanban](#) system, where they:

Guide the identification, success criteria, and prioritization of epics in the funnel and ready states

Warrant reference and consideration in the Lean business case

Influence the definition of the Minimum Viable Product (MVP)

Vision and Solution, ART, and Team Backlogs

Strategic themes influence the vision and backlogs for development at every level. They help determine the values of [Weighted Shortest Job First \(WSJF\)](#) and epic priorities. Solution and [Agile Release Trains](#) ARTs epics that flow from the portfolio, or arise locally, are also influenced by the current strategic themes. Due to their importance, they will often be presented repeatedly by the [Business Owners](#) during [PI Planning](#). Moreover, strategic themes provide vital conceptual alignment across ARTs and [Solution Trains](#).

Value Stream Budgets and Guardrails

Strategic themes profoundly influence value stream [Budgets](#) and [Guardrails](#), which determine the investment and allocation of people needed to accomplish the strategic intent. They trigger the following kinds of questions:

- Do the current value stream investments reflect changes to the business context?
- Are the current value streams the right ones? Should we create a new value stream or decommission an existing one?
- Do the guardrails provide the proper guidance for investments?
- Is there sufficient funding for operations, infrastructure, maintenance, and support activities?
- Are solution investments aligned with the time horizon guardrails to prevent starving the future by over-investing in today or missing near-term opportunities while pouring too much money into the future?

Defining Strategic Themes

Strategic themes are a vital tool for communicating aspects of the strategy to the entire portfolio. LPM can define strategic themes with a simple phrase or using the preferred approach of [Objectives and Key Results \(OKRs\)](#). These themes should be a simple, memorable message that influences everyone involved in solution delivery.

Below are a few examples of strategic themes objectives:

- Appeal to a younger demographic (retail clothing manufacturer)
- Cloud and mobile-first (a financial institution)
- Implement product support for trading foreign securities (a securities company)
- Lower warehouse costs (an online retailer)

Implement single sign-on across applications (independent software vendor)

OKRs help Øprovides a simple approach to create alignment and engagement around measurable and ambitious goals.Ó [2]. Figure 4 provides an OKR example.

Figure 4. Applying OKRs to define a strategic themes

Figure 4. Applying OKRs to define a strategic themes

There are two primary fields in the OKR template:

Objective Ø a memorable description of what the portfolio wants to achieve. These objectives should be short, inspirational, and challenging.

Key results Ø are measurable success criteria for tracking progress towards the objective. Each strategic theme should have two to five key results.

The goal of using OKRs for strategic themes is to define and track their progress through concrete, specific, and *measurable* actions. ØOKRs are frequently set, tracked, and re-evaluated, usually quarterly. OKRs exist to create *alignment* and to set the cadence for the organization. The goal is to ensure everyone is going in the same direction, with clear priorities, in a constant rhythm.Ó [1]

Measuring Progress against Strategic Themes

When strategic themes are defined using OKRs, it offers a way to continually measure progress, allowing the organization to take the necessary corrective actions or amplify successes.

Each *key result* is measurable and should be gradable on an explicit scale, typically a percentage. However, because changes in strategy take time to apply, a quarterly or monthly cadence for measuring and reporting against the key results is recommended.

The Strategic Portfolio Review is the ideal event to reflect on the progress of themes. Figure 5 shows a set of OKRs measured quarterly.

Figure 5. Measuring progress of strategic themes using OKR metrics

Figure 5. Measuring progress of strategic themes using OKR metrics

Learn More

[1] Castro, Felipe. *The Beginner's Guide to OKRs.* <https://felipecastro.com/en/okr/what-is-okr/>

Last update: 14 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



A strategic inflection point is that moment when some combination of technological innovation, market evolution, and customer perception requires the company to make a radical shift or die.

Andy Grove, Only the Paranoid Survive

[Adjust Cookie Settings](#)

Enterprise

The Enterprise represents the business entity to which each SAFe portfolio belongs.

A SAFe portfolio contains one or more [Development Value Streams](#), each dedicated to building, deploying, and supporting a set of [Solutions](#) the enterprise needs to accomplish its business mission. One SAFe [Portfolio](#) can typically govern the entire solution set in small-to-midsize enterprises. In larger organizations—usually those with 500 to 1,000 practitioners or more—multiple SAFe portfolios are needed, often one for each line of business or structured in another way around the business organization and its operating model. In either case, the portfolio is not the entire business. Enterprise and portfolio stakeholders need to ensure that the solutions in each portfolio evolve to meet the broader strategy of the enterprise. This capability is critical to [Business Agility](#).

Details

SAFe's primary focus is helping people build the world's most important systems so do faster and better. Defining that strategy, deciding how much to invest in the solutions, and driving successful execution are critical for every business. This article describes the collaborations and interactions needed between enterprise and portfolio stakeholders to formulate strategy, determine budget allocations, and implement larger enterprise initiatives.

Portfolios within the Enterprise

Smaller enterprises and government agencies may only need a single SAFe portfolio, which can build all the digitally-enabled solutions required to fulfill their mission. The portfolio is connected to the enterprise strategy via portfolio [Strategic Themes](#) and given an approved budget. Enterprise and portfolio stakeholders collaborate to establish these elements, as Figure 1 illustrates.

Figure 1. Small enterprises or government agencies may have only a single SAFe portfolio

Many of the world's largest organizations use SAFe. These enterprises have thousands, and even tens of thousands, of IT, system, application, and solution development practitioners. Not all these practitioners work on the same solutions or development value streams. It's more likely that IT and development personnel are organized to support various lines of business, internal departments, customer segments, or specific business capabilities. In this case, each portfolio is connected to the enterprise as previously described but with three additional considerations, as highlighted in Figure 2 and described below.

Figure 2. A large enterprise may have multiple SAFe portfolios

Figure 2. A large enterprise may have multiple SAFe portfolios

1. Enterprise executives and stakeholders decide how best to allocate the total investment in solutions *across* various portfolios.
2. The enterprise strategy is transformed into a set of strategic themes *for each portfolio*.
3. Some enterprise initiatives (enterprise epics) are often cross-cutting and affect more than one portfolio. Therefore, the enterprise may implement a Kanban system to visualize the flow of these epics.

Each of the critical elements of Figures 1 and 2 is described in the sections below.

Enterprise Strategy Formulation

It all starts with enterprise strategy — a plan of action to achieve the enterprise's mission. It should answer four critical questions about the business:

1. What customers and markets do we serve?

2. What products and solutions do we provide?
3. What unique value and resources do we bring to the endeavor?
4. How will we extend these in the future?

Each enterprise must have an approach to determine its strategy, typically a natural *output* of a logical and reasoned business process. Rather than leaping to conclusions or allowing the loudest or highest-ranking voice to mandate the path, a better approach is first to collaborate and reason about the strategy *inputs*. That will generally lead to agreement and alignment about the strategy. Jim Collins described one such model in *Beyond Entrepreneurship* [1]. Figure 3 adapts Collins's model, highlights the strategic inputs, and defines two SAFe-specific outputs—*portfolio budgets* and *strategic themes*—which link enterprise strategy to portfolio execution.

Figure 3. A model for enterprise strategy formulation

Figure 3. A model for enterprise strategy formulation

Each input to strategy is discussed briefly below:

Vision ☰ represents the future state of what the business wants to be. It's persistent and long-lasting.

Mission ☰ identifies the *current* business objectives that implement the enterprise vision and frame the strategy.

Core values ☰ provide the belief system that governs all individual and corporate behaviors and activities.

Enterprise business drivers ☰ reflect the emerging industry themes and trends that are affecting the business.

Distinctive competence ☰ Strategy naturally leverages the unique advantages that differentiate this business from others and provides a competitive edge.

Financial goals ☰ Whether measured in revenue, profitability, market share, or other metrics, financial or other performance goals are a clear strategy input.

Competitive environment ☰ Competitive analysis identifies the most significant competitive threats to the business.

Portfolio context ☰ Knowledge of the current state of each portfolio informs enterprise strategy. This information is a critical element of enterprise strategy formulation and is described below.

Portfolio Context Informs Enterprise Strategy

Generally, strategic decisions are mostly centralized since they have far-reaching impacts and are often outside the scope, knowledge, and responsibilities of the teams doing the implementation work. In addition, business executives and leaders have the ultimate accountability for business outcomes, so they must be ultimately responsible for the strategy.

However, many factors that inform potential strategy may not be visible to those enterprise executives. Many business challenges, market opportunities, and conditions may be local to various solutions. Consequently, strategy formulation requires continuous collaboration, communication, and alignment with downstream portfolios. In other words, developing an effective strategy demands awareness of the *portfolio context*. This context includes the [Portfolio Vision](#) and [Lean Budget Guardrails](#) that govern the portfolio investments and business [Metrics](#), including outcomes, flow, and competency, as illustrated in Figure 4.



Figure 4. Portfolio context is an essential element of enterprise strategy formulation

Enterprise Portfolio Collaboration

Collaboration between the enterprise and portfolio stakeholders is critical to achieving business goals. While typically led by the most senior enterprise business and technology stakeholders, the process includes participation from portfolio stakeholders who bring important context from their respective value streams, as Figure 5 illustrates.

Figure 5. Stakeholders in the enterprise-portfolio collaboration

Participants in this critical collaboration include:

Enterprise executives with the ultimate responsibility for business outcomes. Some portfolio stakeholders play a significant role in the portfolio vision and enterprise strategy. These can include business owners with primary business and technical responsibility for ROI, including architects (enterprise, system, and solution) and members of the Value Management Office (VMO) that support successful execution and operational excellence.

Epic Owners are crucial in moving large initiatives through and across portfolios.

Enterprise Architects help ensure the strategy is based on feasible technological innovations. In addition, awareness of innovations often provides new strategic opportunities.

Connecting Enterprise Strategy to the Portfolio with Strategic Themes

Strategy formulation is one of the most complex and critical endeavors. SAFe recommends using strategic themes as a summary artifact to communicate strategic intent. When an organization has only one portfolio, a single set of strategic themes is sufficient. In contrast, when an organization has more than one portfolio, a set of strategic themes for the enterprise may be needed to inform and

connect to each portfolio's strategy, as illustrated in Figure 6.

Figure 6. Enterprise and portfolio strategic themes

Figure 6. Enterprise and portfolio strategic themes

A typical format for a strategic theme is a short, simple phrase (for example, Expand to the European market, Transition to the cloud, Enable consumer self-service). A more rigorous approach uses [OKRs \(Objectives and Key Results\)](#). Either way, strategic themes communicate strategic intent to everyone in the organization.

Allocating Portfolio Budgets

An effective lean enterprise dynamically allocates funds across portfolios to execute the evolving strategy. To support this flexibility and speed of decision-

making, rather than funding individual projects, SAFe enterprises allocate budgets to each portfolio, which is empowered to make investments that maximize the value returned, as illustrated in Figure 7.

Figure 7. Portfolio budget allocations

Figure 7. Portfolio budget allocations

As the [Lean Budgets](#) article describes, each portfolio, in turn allocates budgets to its development value streams.

Managing Cross-Portfolio Initiatives

Solutions are generally managed within a specific portfolio. Some initiatives, however, require the collaboration of multiple portfolios (for example, implementing GDPR compliance across all solutions in the enterprise). Many are critical and not subject to debate (like the GDPR example). And yet, if not managed appropriately, these organization-wide initiatives can be blindly pushed onto busy

portfolios and overload the system.

Enterprise epics can be created to define and reason about this significant work. Like the [Portfolio Kanban](#), the work intake process for the enterprise Kanban helps match demand to capacity, preventing overload and fostering faster value delivery. Figure 8 illustrates an example of an enterprise epic kanban system.

Figure 8. An enterprise epic Kanban

Figure 8. An enterprise epic Kanban

The process can be summarized as follows:

All significant new cross-portfolio initiatives enter the Funnel of the Kanban and progress through Review and Analysis in close collaboration with the portfolios that will do the work.

Like portfolio epics, the epic hypothesis statement and the lean business case can be used to define and further elaborate enterprise epics.

After the analysis is complete, a go/no-go decision is made.

Approved enterprise epics are pulled into implementation by portfolios that create

epics to describe their portion of the work.

As with the portfolio, enterprise Epic Owners foster and drive collaboration around cross-portfolio initiatives. In many cases, the scope and implementation rhythm of the corresponding portfolio epics may require synchronization across the enterprise. For example, an enterprise epic may require a coordinated MVP (Minimal Viable Product) to validate the business hypothesis (Figure 9). This limits investment risk and allows exploratory discovery for the most significant and critical enterprise initiatives (see the SAFe Lean Startup Cycle in [Epics](#)).

Figure 9. Coordinating an MVP across multiple portfolios

Figure 9. Coordinating an MVP across multiple portfolios

Capturing Strategy in a Business Model Canvas

The concepts outlined in this article provide a logical and reasoned process in which enterprise strategy formulation reflects internal and external objectives,

business conditions, and the organization's larger purpose. Moreover, the plan must be communicated and made clear to all stakeholders. There is, of course, no one right way to do this. One popular way to frame a strategic plan is through the Business Model Canvas [2]. The BMC is a one-page template that summarizes the most important aspects of a business model, as illustrated in Figure 10.

Figure 10. The Business Model Canvas

[Download The Business Model Canvas](#)

The BMC can model any business, from a startup to a global enterprise. It has nine independent building blocks that help clarify thinking and focus when describing a business model. These blocks are defined as follows:

Value propositions are the set of products and services the enterprise offers.

Key partners are the various buyer-supplier relationships and business alliances that facilitate achieving the value proposition.

Key activities are the enterprise's most important actions to deliver its products and services.

Key resources are the critical human, physical, intellectual, financial, and other capabilities and assets the enterprise has to achieve its objectives.

Customer relationships define the type of relationships needed to apply and leverage the business's products and services.

Customer segments define how the business views and treats various sets of customers differently based on their common attributes.

Channels explain how the enterprise delivers its products and services to intermediaries, customers, and end-users.

Cost structure highlights the various development, production, deployment, operating, and support costs associated with the business's products and services.

Revenue streams reflect how the enterprise receives financial compensation for its products and services.

Thinking Like a Lean Startup with the Lean Canvas

A popular variant of the Business Model Canvas is the *Lean Canvas* created by Ash Maurya [3], illustrated in Figure 11 [4].

Figure 11. The Lean Canvas

Figure 11. The Lean Canvas

[Download The Lean Canvas](#)

The Lean Canvas is derived from the thinking in Lean Startup [4], designed to address the unique opportunities and challenges in startup enterprises. It's now also applied to innovation efforts in larger enterprises. The Lean Canvas is similar to the BMC. However, it focuses more on the problem to be solved and the unique enterprise competencies to address emerging opportunities.

The Lean Canvas has nine blocks. Four blocks are the same as the BMC: channels, customer segments, revenue streams, and cost structure. However, the Lean Canvas replaces the other five blocks with the following items:

Problem describes the *problem* in the marketplace before designing a potential solution, a signature aspect of [Design Thinking](#).

Solution defines the key characteristics of a proposed solution.

Unique Value Proposition describes the unique skills, resources, and assets a business brings to the endeavor. In other words, "why you are different and worth getting attention." [3]

Unfair advantage describes the tangible and intangible assets of the business that cannot be easily bought, copied, or replicated by competitors who address the same problem.

Key metrics identify the critical, early measures indicating whether the new solution will likely address the problem.

The Lean Canvas helps define an actionable business plan. It focuses on customer problems, solutions, key metrics, and competitive advantages. In contrast, the Business Model Canvas is a strategic management tool allowing you to describe, design, challenge, invent, and pivot your existing business model [5]. While neither canvas captures all the elements of an enterprise strategy, both are valuable tools to evolve the organization's solution portfolios.

Learn More

[1] Collins, Jim, and Lazier, William. *Beyond Entrepreneurship: Turning Your Business into a Great and Enduring Company*. Prentice-Hall, 1992.

[2] Osterwalder, Alexander, Yves Pigneur, and Tim Clark. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons, 2010.

[3] Maurya, Ash. *Running Lean: Iterate from Plan A to a Plan That Works*. O'Reilly Media, 2012.

[4] The Lean Canvas. <https://leanstack.com/leancanvas>

[5] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Random House, Inc, 2011.

[6] <https://www.eqengineered.com/insights/why-use-lean-vs-business-model-canvas>

Last update: 11 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

English

Scaled Agile Framework



*A specialist is a man
who knows more
and more about
less and less.*

~William J. Mayo

Shared Services

Shared Services represents the specialty roles, people, and services required for the success of an ART or Solution Train, but that are not dedicated full-time.

[Adjust Cookie Settings](#)

Because these individuals have specialized skills—often single-sourced and typically

quite busy—each [Agile Release Train](#) (ART) and [Solution Train](#) must plan to engage the shared services personnel it needs when it needs them.

Details

ARTs and, by extension, Solution Trains assemble all the necessary knowledge, skills, and abilities needed to deliver value. Without it, teams can be blocked waiting for information or decisions that delay value delivery. However, it's impossible to devote some specialty functions to a single ART. There may be a shortage of a particular skill available or it is simply impractical to staff all ARTs with that expertise. Also, the needs of the ART may fluctuate, making full-time commitment impractical. To address this, Shared Services support development by quickly focusing specialty expertise on the areas of the system or [Solution](#) that require unique knowledge and skills.

In some cases, the effort may contribute directly to the [Architectural Runway](#) (for example, security and information architecture) that supports new Feature or [Capability](#) development. In others, the effort can trail core development a bit (for example, customer training, localizations, and compliance audits). In some cases, merely being supportive and reactive quickly is sufficient. These services on the trains ensure their work is performed early and in smaller batches to make delivery more predictable and provide team feedback.

In either case, the programs will struggle to meet their objectives without timely support and synchronization. Shared Services may participate full-time with a single ART or distribute their efforts across multiple ARTs across the [Enterprise](#). And in some instances, they may embed themselves directly on a single [Agile Team](#) for a short time. In all cases, they commit with the teams and ARTs to the shared goals and [PI Objectives](#).

Shared Services often include people with the following types of specialized skills:

Application/web portal management	Infrastructure and tools management
Site reliability engineering	Internationalization and localization support
DevOps specialists	IT Service Management (ITSM) and deployment operations
Configuration management	Security specialist (InfoSec)
Data modeling, data engineering, and database support	Regulatory and compliance
End-user training	System QA and exploratory testing
Enterprise architecture	Technical writers
Information architecture	

Responsibilities

Depending on the type of service they provide, Shared Services personnel may engage in the following type of activities:

- Participating in [PI Planning](#) and [Pre-PI Planning](#) for Solution Trains.
- Contributing requirements where necessary, adding to [Solution Intent](#) and taking ownership of their portion of dependent backlog items.
- Collaborating with Agile teams to fulfill the dependencies that occur during PI execution.
- Participating in [System Demos](#), [Solution Demos](#), and [Inspect and Adapt \(I&A\)](#) workshops, when appropriate, as many improvement backlog items may reflect challenges with the availability of specialized skills and dependencies.

Occasionally, members of Shared Services may choose to operate as a single team. In this case, they would iterate on the same cadence as the ARTs and work like any other Agile team.

Patterns of Shared Services

Beyond the generalized purpose and responsibilities of shared services in SAFe, there are multiple patterns for how Shared Services are deployed in the organization, depending on the context. The following paragraphs describe three common patterns observed in the field.

Centralized Services

The most frequently observed use of Shared Services is when one or more individuals with specialized skills from a functional department provide periodic support to an ART. For example, if an ART needs to acquire products or services from a supplier, they will likely need focused help from procurement, legal, and accounting. Launching a new or updated product will require various specialized skills from the marketing department to support the release process. Even highly specialized technical skills such as enterprise data architecture may be managed centrally, and support provided to ARTs as a Shared Service.

Embedding in Agile Teams

Shared Services personnel may temporarily become part of an Agile team for short periods. In doing so, they experience the collaborative, quick, and high-quality way Agile teams produce products. It also accelerates the larger teams-of-Agile-teams dynamic that—only by acting together—can they deliver enterprise value.

Embedding also transfers knowledge rapidly and in real-time, reducing delays and dependencies on specialized skills. For example, one government customer brought in front-line personnel from the field on six-month temporary duty assignments to serve as embedded Product Owners in the Agile teams. That action accelerated the development of the next-generation software that those agents would use when the solution was released.

Shared Services as Enabling Teams

In many cases, Shared Services personnel may operate perpetually as a shared service due to the nature of their specialized services. However, as Agile Teams grow and mature, and as responsibility for delivery across the whole value stream moves to teams and trains, shared services may appear as perpetual dependencies and bottlenecks to flow.

Another approach is configuring Shared Services teams as enablement teams (see applying Team Topologies in SAFe). As such, they are chartered with teaching and enabling Agile teams and ARTs the specialized technologies, services, and practices they have been providing. One example of this pattern is teams of site reliability engineers (SREs) equipping Agile teams with the skills and practices necessary to assure site reliability themselves.

Mastering this approach is a hallmark of business agility as it moves authority and responsibility for full value stream product and solution delivery to the teams and trains who deserve that charter.

Tips for Successful Application of Shared Services

The following is a short list of tips for successfully applying the concept of Shared Services in SAFe enterprises.

Use Shared Services when specific skills are truly not needed on an ART full-time (if the skill is required full-time, staff the ART accordingly)

Ensure Shared Services is a full participant in ART events, working agreements, DoD, standards, and norms

Encourage Shared Services members to share their expertise freely to build T-shaped skills on the Agile teams

Share wins with other Shared Services to support their relentless improvement efforts

Prioritize Shared Services contributions to unblock value streams

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 12 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



It's said that a wise person learns from his mistakes. A wiser one learns from others' mistakes. But the wisest person of all learns from others' successes.

—Zen proverb
adapted by John C. Maxwell

[Adjust Cookie Settings](#)

Communities of Practice

Communities of Practice (CoPs) are organized groups of people with a common interest in a specific technical or business domain. They regularly collaborate to share information, improve their skills, and actively work on advancing their knowledge of the domain.

Healthy CoPs have a culture built on professional networking, personal relationships, shared knowledge, and common skills. Combined with voluntary participation, CoPs provide knowledge workers with opportunities to experience autonomy, mastery, and purpose beyond their daily tasks on an [Agile Release Train \(ART\)](#) [2].

CoPs enable SAFe participants to exchange knowledge and skills with people across the entire organization. This open membership offers access to a wide range of expertise to help with technical challenges, fuel continuous improvement. It allows more meaningful contributions to the larger goals of the [Enterprise](#). The result is that organizations benefit from rapid problem-solving, improved quality, cooperation across multiple domains, and increased retention of top talent.

Details

According to Wenger [1], CoPs must have three distinct traits to be considered a community of practice, as shown in Figure 1.

Domain □ An area of shared interest

Practice □ A shared body of knowledge, experiences, and techniques

Community □ A self-selected group of individuals who care enough about the topic to participate in regular interactions

Figure 1. Communities of Practice have three distinct traits

Figure 1. Communities of Practice have three distinct traits

Lean-Agile principles and practices promote cross-functional teams and programs that facilitate value delivery in the enterprise. Similarly, Lean thinking emphasizes organizing people cross-functionally with different skills around a value stream. However, software engineers need to talk with other software engineers; data scientists need to talk with other data scientists; Product Owners need to communicate with their peers from other [Agile Teams](#), and so on. This is critical for leveraging the multiple experiences and different types of practical knowledge available from various people. That drives craftsmanship and continuous learning (see the [Continuous Learning Culture](#) competency article), facilitating the adoption of new methods and techniques.

Such domain-focused interactions are often supported by CoPs—informal networks designed specifically for efficient knowledge-sharing and exploration across teams, trains, and the entire organization. Figure 2 provides an example of role-based CoPs, one of the most common types of communities.

Figure 2. Role-based Communities of Practice

Figure 2. Role-based Communities of Practice

For example, [Scrum Masters/Team Coaches](#) from different Agile teams may form a CoP to exchange practices and experiences in building highly productive Agile teams. As CoPs start to gain acceptance and participation, topic-based communities like those shown in Figure 3 often emerge.

Figure 3. Topic-based Communities of Practice

Figure 3. Topic-based Communities of Practice

The membership of these CoPs can be far more diverse. A CoP on [DevOps](#) could attract participants from almost any role in an organization.

A CoP with [Product Owners](#) and [Business Owners](#) could focus on how to write better features optimized for improving the flow of value through the ART. Another CoP could be formed to improve the organization's ability to measure flow by building systems to capture and report [flow metrics](#). Software engineers may form a CoP to help members cross-train on low code development platforms when the company adopts that technology as a future strategic direction. Leaders from ARTs, Solution Trains, and the Portfolio may form a CoP to better understand [Value Stream Management](#) and build the practices their enterprise will follow to better manage the [Development Value Streams](#).

Organizing a Community of Practice

CoPs are highly organic, and like most living organisms, they have a natural life cycle, beginning with an idea for a new community and ending when the community members feel the group has achieved its objectives or is no longer providing value. Figure 4 shows the typical life cycle of a CoP.

Figure 4. CoPs typically follow a five-stage life cycle, from conceptualization to closure
(3)

CoPs are formed in the committing stage by a small, core group of practitioners who share a common passion and need for a particular domain. As shown in Figure 5, CoP members exhibit multiple levels of participation.

Figure 5. CoP members exhibit multiple levels of participation and can move freely across the levels as needs and interests evolve.

Each level is described next:

Core team ▷ The core team forms the heart of the community that will organize, charter, market, nurture, and operate the community.

Active ▷ These members work closely with the core team to help shape the definition and direction of the CoP. This includes defining the community's shared vision, purpose, roles, strategies for interaction, marketing, and communications.

Occasional ▷ These members participate when specific topics of interest are addressed or when they have something to contribute to the group. They are often the largest group in the community.

Peripheral ▷ These members feel connected to the community but engage on a limited basis. These could be newcomers or those more interested in community activities.

Transactional ▷ These members are the least connected to the community. They may connect only to access CoP resources or to provide a specific service to the CoP (for example, website support).

It's common for people to move between different levels of participation and commitment over time. CoPs are self-organizing, and their members have the freedom to determine their level of engagement that is different from other work groups, such as tiger teams, task forces, and committees. The natural movement of people among communities and levels is healthy. It allows new knowledge and fresh ideas to flow across the organization in ways that are different but complementary to formal information-sharing.

Operating a Community of Practice

Since CoPs are informal and self-managing by nature, community members are empowered to design the types of interactions and determine the frequency that best meets their needs. For developers, this could involve hackathons, coding dojos, and tech talks. Other formats might include meetups, brown bags, webinars, and independent communications through social business platforms such as Slack, Confluence, and Jive.

In the operating stage of a CoP, community members continuously evolve by engaging in periodic retrospectives similar to those used by Agile teams. Core team members focus on maintaining the health of the community by:

- Keeping things simple and informal
- Fostering trust
- Ensuring the rapid flow of communication and shared awareness
- Increasing the shared body of knowledge developed in the CoP

Eventually, individual CoPs will run their course, and community members should consider retiring the CoP, allowing practitioners to commit their energies to other communities. Signals that a community has reached this stage include a steady decline in event participation, reduced activity on collaboration sites, and input from community retrospectives. When a CoP is retired, leaders should make it a positive event where community successes are celebrated, key contributors are recognized, and ongoing participation in other CoPs is encouraged. CoP experiences often become part of company lore through these celebrations, and it is not uncommon for a healthy CoP retirement to spawn three to five new communities.

Fostering Engagement in Communities of Practice

The [Innovation and Planning \(IP\) Iteration](#) presents an excellent opportunity for CoPs to hold learning sessions, formal or informal, as well as other activities such as coding dojos, coaching clinics, and the like.

The role of [Lean-Agile Leaders](#) is to encourage and support people's desire to improve. This helps the enterprise improve and unlocks the intrinsic motivation of knowledge workers, as discussed in [SAFe Principle #8 - Unlock the intrinsic motivation of knowledge workers](#). CoPs embrace the ideals of *alignment, transparency,*

respect for people, and *relentless improvement*, described in the [SAFe Core Values](#).

By fostering CoP formation, [Lean-Agile](#) leaders show support by continuously communicating the value of CoPs, highlighting success stories, and recognizing the efforts of community volunteers. Leaders can also support CoPs by providing meeting spaces, logistical support, and funding for meetups, tooling, and communications infrastructure.

Learn More

[1] Wenger, Etienne. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1999.

[2] Pink, Daniel H. *Drive: The Surprising Truth About What Motivates Us*. Riverhead Books, 2009.

[3] The Distance Consulting Company. *Community of Practice Start-Up Kit*. 2000.

Last update: 18 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite

SAFe videos and
presentations

Blog

TRAINING

Course Calendar

About Certification

Become a Trainer

GET SOCIAL

Twitter

Linkedin

YouTube

Permissions Form

Usage and Permissions

300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Prediction is very difficult, especially if it is about the future.

—Niels Bohr, Danish physicist

Roadmap

The Roadmap is a schedule of events and milestones that forecasts and communicates planned solution deliverables over a time horizon.

[Adjust Cookie Settings](#)

Roadmaps are a visual tool that assists in the development and communication of

planned deliverables, milestones, and investments over time and help distinguish different types of work. Roadmaps are the glue that links strategy to execution and offer the ability to develop, evolve and adjust planned activities. They also provide stakeholders with a view of the current, near-term, and longer-term deliverables that realize some portion of the [Portfolio Vision](#) and [Strategic Themes](#).

Details

Responding to change over following a plan is one of the four values of the Agile Manifesto [1]. While this value emphasizes responding to change, it assumes that there is a plan to follow. Indeed, planning and road mapping are fundamental to Agile.

There are three types of roadmaps defined in SAFe:

1. **PI roadmap** D Illustrates commitments for an [Agile Release Train \(ART\)](#) or [Solution Train](#) for the planned, upcoming PI. The forecast may provide the deliverables and milestones for the following two PIs.
2. **Solution roadmap** D Provides a *longer-term* often multiyear view showing the key milestones and deliverables needed to achieve the solution [Vision](#) over time.
3. **Portfolio roadmap** D Shows an *aggregated multiyear* view of how LPM will achieve the portfolio vision across all its [Value Streams](#).

While forecasts for innovation and invention are inherently uncertain, organizations still need roadmaps for many reasons:

Steering significant initiatives D Some initiatives may take years to develop, requiring a longer horizon for planning and coordinating activities and deliverables to achieve the business objectives.

Preparing for releases D [Customers](#), [Suppliers](#), and partners need to understand how [Solutions](#) might be implemented and evolve and how they will achieve the vision. Customers also need time to plan for changes and how to test and implement the solution in their environment.

Addressing strategic concerns D [Government](#) agencies and others periodically publish new regulations. Ensuring [Compliance](#) is a strategic concern; road mapping can help visualize and track the necessary deliverables and milestones.

Aligning stakeholders D Internal stakeholders such as finance, sales, and marketing need time to align with the development organization to establish financial forecasts, build and execute sales and marketing campaigns, and communicate with partners and Customers.

Milestones are an essential element of roadmaps as they mark specific progress points on the development timeline and are critical for understanding and

monitoring product evolution and risk. SAFe defines the following types of milestones:

PIs are time-based milestones that appear on most roadmaps as they provide cadence-based, objective measures of progress

Fixed-date milestones include events, release dates, contractual obligations, and scheduling of deliverables that must occur on or before a specific date.

Learning milestones help validate technical and business opportunities and hypotheses.

Roadmaps, including milestones, provide stakeholders with a means to understand, collaboratively shape, and plan for future solutions, helping the enterprise, Customers, and Suppliers achieve the desired outcomes.

Applying Planning Horizons

Creating effective roadmaps requires an understanding of the appropriate time horizon. If the planning horizon is too short, the enterprise may jeopardize alignment and the ability to communicate new future [Features and Capabilities](#). Too long, the enterprise is basing assumptions and implying commitments on an uncertain and distant future. Multiple planning horizons provide a proper balance (Figure 1).

Figure 1. SAFe planning horizons

Figure 1. SAFe planning horizons

The outer levels of the planning horizon are longer-term and describe less defined and less committed behavior. In contrast, the inner levels are nearer-term, defining better understood and more committed solution behavior. The following sections describe each planning horizon.

Daily Plan

The daily plan (*team sync*) is a short meeting (usually 15 minutes or less), typically held daily, to coordinate the team's work, inspect progress toward the iteration goal, communicate, and adjust upcoming planned work.

Iteration Plan

In SAFe Scrum, [Iteration Planning](#) is a structured event that kicks off each iteration.

Teams collaborate to determine how much of the team backlog they can deliver during an upcoming iteration and summarize those [Stories](#) into a set of [Iteration Goals](#) and the iteration backlog. That is their ŒplanŒ and commitment to the ART business.

SAFe Kanban Teams often follow a similar pattern as Scrum, participating in each ART iteration, contributing to System Demos, implementing stories to advance toward their PI Objectives, and collaborating with other teams. However, Kanban teams do not have to plan at iteration boundaries, maintain an iteration backlog, or establish iteration goals. They complete their work within a timebox required to meet business and technical needs and class of service (e.g., fixed date, expedite). Kanban teams ensure they have a sufficient amount of prioritized backlog items to keep them productive and avoid delays. That is their ŒplanŒ and commitment to the ART business.

PI Plan

The PI Plan is the output of the ARTŒs most recent [PI Planning](#) event. During PI planning, Agile Teams get presented with new [Features](#) and plan the stories they need to deliver alongside work from their local context. Teams summarize this work as a set of team PI Objectives. The PI plan also includes the identification of risks and the ART planning board, which illustrates how the features will be delivered over time and highlight any dependencies.

PI Roadmap

The *PI roadmap* typically consists of a summary of the PI Plan plus 1-2 *forecasted* PIs. The current PI is where the ART committed to the [PI Objectives](#). Since the business and technical context may change, planning subsequent PIs is less precise and defined. ARTs can generally predict with relatively good confidence since they use their historical average velocity.

Product Management generally leaves room for new [Features](#) in the following PIs, scheduling the amount of work to be less than available capacity. Each element on the roadmap is a feature, [Capability](#) (or ART [Epic](#)), intended for a particular PI. The PI roadmap may also reflect fixed-date and learning milestones during that period.

Figure 2 illustrates a roadmap that consists of a summarized *PI Plan* and two *forecasted* PIs. This timeframe is typically sufficient to communicate intent with stakeholders, the business, and partners. ItŒs also short enough to keep commitments from interfering with adjustments to changing business priorities.

Figure 2. An example of a PI roadmap for an autonomous vehicle

Figure 2. An example of a PI roadmap for an autonomous vehicle

Figure 2 also illustrates that some organizations prefer to create PI roadmaps with less than their available capacity, creating a more opportunistic plan for change. Others plan future PIs with a more complete but riskier schedule. Regardless of how PIs are forecast, items don't become committed until teams plan them during PI planning.

The Solution Roadmap

The *Solution Roadmap* in Figure 3 provides a multiyear view of planned epics and capabilities time for a specific *solution* and timeframe within the portfolio.

Figure 3. The solution roadmap for an autonomous delivery vehicle

Figure 3. The solution roadmap for an autonomous delivery vehicle

In this example, the roadmap depicts milestones as diamonds, while a solution release is shown with a small box. As the time horizon extends, items are less defined and more uncertain. For example, the first year is planned in quarters (Q1, Q2, Q3, and Q4), which may or may not align with PI boundaries. The second year is shown in six-month increments (H1 and H2). Anything beyond that is scheduled in years (Y3 and Y4) to reflect the significantly higher uncertainty.

The Portfolio Roadmap

The [Portfolio Roadmap](#) (Figure 4) illustrates the plan of intent for achieving the portfolio vision, primarily with epics, for an extended time.

Figure 4. The portfolio roadmap communicates the longer-term vision

Figure 4. The portfolio roadmap communicates the longer-term vision

The portfolio roadmap integrates the various aspects of solution and PI roadmaps and their milestones into a comprehensive view across all the value streams in the portfolio. This roadmap builds the larger picture, communicating how the portfolio vision will be achieved over a specific timeframe to stakeholders and illustrates a high-level view of [Epics](#) within each value stream.

Since the solution and portfolio roadmap may span multiple years, both require estimating longer-term initiatives. However, every enterprise must be cautious about such forecasts. While many see long-term predictability as a goal, [Lean-Agile Leaders](#) know that *every long-term commitment decreases the agility of the enterprise*. If the future is too fixed, it's challenging to have [Business Agility](#). Therefore, regular forecast updates reflect new learning and changing market conditions.

Flexible Roadmaps Improve Flow

Customer desires, competitive threats, technology choices, business expectations, revenue opportunities, and workforce demands now happen at *blistering speeds*. Consequently, roadmaps need to respond to change and be flexible.

Principle #6, Make value flow without interruptions, informs us that working in smaller batches reduces queue lengths. Committing to longer-term roadmaps often creates queues. The math in Little's Law tells us that longer queues increase the wait time for any new initiative.

One way to ensure a flexible roadmap is to focus on high-value items and limit longer-term fixed-date commitments. Let's explore two different scenarios to illustrate this situation:

Partially loaded PI roadmap ▷ Figure 2 above shows a PI roadmap with room for new features in the following PI. If an item can't fit in the current PI, it can be scheduled for the next, which is approximately ten weeks, depending on the chosen PI duration.

Fully loaded PI Roadmap ▷ Figure 5 below illustrates a roadmap where all five PIs are fully loaded and committed. In this case, the wait time for a new feature is more than 50 weeks (assuming a ten-week PI duration). Shifting from a traditional to an Agile mindset means keeping commitments as short and flexible as possible.

Figure 5. A fully committed roadmap becomes a queue, increasing the wait time

Figure 5. A fully committed roadmap becomes a queue, increasing the wait time

Milestones Make Critical Events Visible

Internal and external milestones, and significant events, offer critical insights into building roadmaps [2]. They help stakeholders understand feature progress and visualize and leverage opportunities that can be predictable and require longer-term planning, as described in the following sections.

Milestones mark specific progress points on the development timeline and are critical for understanding and monitoring product evolution and risk. In the past, many progress milestones were based on phase-gate activities. But experience has shown that stage gate milestones generally do not reduce risk. (See [Principle #5 – Base milestones on objective evidence of working solutions.](#))

SAFe progress milestones are represented by the fixed cadence of Iterations and **PIs**, as illustrated in Figure 6. It defines three types of milestones:

1. **PI milestones** ☰ Each PI creates an objective measure of progress, as shown in Figure 1. These milestones help evaluate progress toward the technical or business hypothesis.

Figure 6. PI milestones provide objective evidence

Figure 6. PI milestones provide objective evidence

With SAFe's approach, the system is built in increments, each of which is an integration and knowledge point that demonstrates evidence of the solution's viability. Further, these objective evaluations are performed regularly on a PI cadence, which ensures periodic availability and evaluation. It also offers predetermined time boundaries for eliminating less desirable solution options (See [Principle #3, Assume variability; preserve options](#))

1. **Fixed-date milestones** ☰ Every Lean-Agile enterprise wants to operate with the minimum possible constraints, providing more agility. However, fixed-date milestones are typically for traditional and Lean-Agile development. For example, they are required to communicate the following:

Events include trade shows, customer demos, user group meetings, planned product announcements

Release dates for internal or external business concerns

Contracts with binding dates for value delivery, intermediate and payment milestones, demos, and more

Scheduling larger-scale integration, including hardware, software, supplier solutions, and anything else where a fixed date provides an appropriate synchronization to bring together assets and validate them

Fixed-date milestones should be reflected in the relevant roadmaps, so stakeholders can plan and act accordingly.

1. **Learning milestones** help validate business opportunities and hypotheses, as shown in Figure 7.

Figure 7. Learning milestones help evaluate progress toward the goal

Figure 7. Learning milestones help evaluate progress toward the goal

Many other concerns also affect planning and road mapping. Things like patent filing, internal or regulatory audits and certification, user acceptance testing, and more must be identified and made visible on roadmaps.

Market Rhythms and Market Events Influence Roadmaps

Most enterprises operate in a larger ecosystem of consumers, buyers, purchasing cycles, and supply chains affected by external calendar-driven factors and constraints. Understanding market rhythms and events help companies leverage predictable, calendar-based opportunities and require longer-term planning.

Market Rhythms

A *market rhythm* is a set of significant events that occur periodically on a predictable cadence. For example, retailers routinely prepare for the holiday shopping season by upgrading their computer systems to get a competitive edge and support significantly higher transaction volumes. In these cases, release dates can dramatically affect the value delivered.

For example, Figure 8 illustrates the market rhythms for three different companies.

Figure 8. Example market rhythms for different companies

Figure 8. Example market rhythms for different companies

The vertical axis in Figure 8. shows the value delivered to a market, while the horizontal axis shows the value over time, usually a calendar or fiscal year.

1. **B2B retail software** Ⓛ the blue line shows a similar market rhythm as the toy maker. However, this organization must release its solutions even sooner.
2. **B2C social media** Ⓛ the green, flat horizontal line represents a company where the value over time is relatively constant, suggesting that market rhythms do not strongly influence it.
3. **Toy maker** Ⓛ the orange line illustrates an organization that must have its products ready for sale before the annual holiday shopping season.

The *toy maker* and *B2B retail software* companies make most of their sales during the holiday season. In the extreme case, a product or service may have zero value if that market window is missed. This wide range of potential value correlated to the calendar can substantially impact business outcomes. Agile or not, planning is a critical activity.

Market Events

A *market event* is a *one-time* future event with a high probability of materially affecting the value of one or more solutions. Market events can be external, such as the launch of government regulations, or they can be internally created, such as a company's annual user conference. Market events (Figure 9) are typically represented on the roadmap as milestones. They may strongly impact specific solution releases. Consequently, it may cause adjustments to the timing of features or solution development activities, often identified during [PI Planning](#).

Figure 9. Example of market events

Figure 9. Example of market events

Learn More

[1] The Agile Manifesto. <http://agilemanifesto.org>

[2] Hohmann, Luke. *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Addison-Wesley Professional, 2003.

[3] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. Amacom, 2010.

Last update: 13 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

[GET SOCIAL](#)
[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The epiphany of integration points is that they control product development. They are the leverage points to improve the system. When timing of integration points slip, the project is in trouble.

ÑDantar Oosterwal,
The Lean Machine

Adjust Cookie Settings

Continuous Integration

Continuous Integration (CI) is an aspect of the Continuous Delivery Pipeline in which new functionality is developed, tested, integrated, and validated in preparation for deployment and release.

CI is the second aspect in the four-part [Continuous Delivery Pipeline](#) of [Continuous Exploration](#) (CE), Continuous Integration (CI), [Continuous Deployment](#) (CD), and [Release on Demand](#) (Figure 1).

Figure 1. Continuous integration in the context of the continuous delivery pipeline.

Details

Continuous integration is a critical technical practice for each Agile Release Train (ART). It improves quality, reduces risk, and establishes a fast, reliable, and

sustainable development pace.

With continuous integration, the system always runs, meaning it's potentially deployable, even during development. CI is most easily applied to software solutions where small, tested vertical threads can deliver value independently. In larger, multi-platform software systems, the challenge is harder. Each platform has technical constructs which need continuous integration to validate new functionality. CI is even more complicated when systems comprise software, hardware, components, and services provided by suppliers. But the fact remains that frequently integrating and testing features together is the only practical way to validate a solution fully.

As a result, teams need a balanced approach that allows them to build-in quality and gets fast feedback on their integrated work. For purely software-based solutions, continuous integration is relatively easy to achieve with modern tools. For more complex systems with hardware and software, a *continuous integration* approach is required (see the [Enterprise Solution Delivery](#) article) to balance the economic trade-offs between frequency, the scope of integration, and testing.

The Four Activities of Continuous Integration

As illustrated in Figure 2, SAFe describes four activities associated with continuous integration:

1. **Develop** describes the practices necessary to implement stories and commit the code and components to version control
2. **Build** describes the techniques needed to create deployable binaries and merge development branches into the trunk
3. **Test end-to-end** describes the practices necessary to validate the solution
4. **Stage** describes the steps required to host and validate solutions in a staging environment before production

Figure 2. Four activities of continuous integration

Figure 2. Four activities of continuous integration

Develop

Developing the solution refers to implementing stories by refining features from the [ART Backlog](#) as may be needed and then coding, testing, and committing the work product into the source control system. Testing in this activity tends to focus on unit and story-level testing and often requires test doubles (see [Test-Driven Development](#)) to replicate other components or subsystems that are not readily available or easily tested.

Several practices are associated with developing the solution:

Break features into stories → This enables continuous delivery via small batches and smooth integration, including creating user story maps to ensure that workflows meet customer needs.

Behavior-Driven Development (BDD) → BDD is a process Product Owners and teams use to understand requirements better and improve quality by creating acceptance criteria and automating tests, often before the code is written. BDD

works with TDD and is described [here](#).

Test-Driven Development (TDD) \triangleright TDD involves writing the unit test first, then building the minimal code needed to pass the test. This testing technique leads to better design, higher quality, and increased productivity. TDD works with BDD and is described further [here](#).

Version control \triangleright Effective version control allows teams to recover quickly from problems and improve quality, ensuring the integration of the right components. Aggregating assets under version control is a leading indicator of continuous integration maturity.

Built-in quality \triangleright [Built-In Quality](#) prescribes practices around flow, architecture & design quality, code quality, system quality, and release quality.

Application telemetry \triangleright Application telemetry is the primary mechanism that acquires and then uses application data to help determine the results of relevant hypotheses.

Threat modeling \triangleright In addition to threat modeling done in continuous exploration (architect step), the system design should identify possible vulnerabilities that teams may introduce with new functionality.

Build

During the build phase, teams continuously integrate new code. Automating the build and test tools to run upon code commit is one of the best ways to integrate. Passing versus not-yet-passing and broken automated tests are objective indicators of progress. Automating code building enables teams to fix problems quickly *before they affect more significant parts of the system*. Addressing a broken build should be the highest priority. A \triangleright gated commit ensures software has passed the gate (e.g., unit tested, performance-tested, free of known defects, and so on) before being checked into the main codebase or *trunk*. Code that passes the tests is automatically integrated, which removes the complications of managing multiple source code branches. *Trunk-based development* helps ensure code can be released on demand reliably without costly code freezes.

Five practices can help build a high-quality solution:

1. **Continuous code integration** \triangleright Code commit should automatically trigger the compilation and testing of changes. Ideally, this happens on each commit and should happen several times daily.
2. **Build and test automation** \triangleright The automated compilation process includes unit- and story-level tests to verify changes, often requiring *test doubles* to enable fast builds and replicate other systems.
3. **Trunk-based development** \triangleright Teams should integrate code quickly, at least once daily, or ideally upon commit, and all teams should work off a single trunk, avoiding long-lived branches.
4. **Gated commit** \triangleright Committing to the main trunk is risky, as broken changes can impact many teams. Therefore, only the modifications validated through the build

and test process merge into this branch.

5. **Application security** ☐ Code analysis tools inspect the code and third-party packages for known vulnerabilities.

Test the solution end-to-end

While critical, automated local story and component testing aren't enough.

System-level integration and testing are required to test features thoroughly.

Figure 3 illustrates how the [System Team](#) helps integrate the work of all teams on the ART frequently, providing some objective evidence of progress.

Figure 3. Integrating the work of all teams on the ART

Figure 3. Integrating the work of all teams on the ART

System-level testing frequently happens during the iteration, ideally after every commit. However, whatever the circumstances, such full-system integration must be accomplished at least once per iteration. Otherwise, the late discovery of defects and issues from earlier iterations causes substantial rework and delays.

Five practices can help improve end-to-end system testing:

1. **Test and production environment congruity** ⚡ Environment congruity assures that testing exercises the solution as it would behave in front of live users and decreases the probability that defects will escape into production.
2. **Test automation** ⚡ Automated testing should include a variety of tests, such as functional, integration, regression, and more. The [Agile Testing](#) article details a testing matrix of what can and should be automated.
3. **Test data management** ⚡ To create stability, tests must be consistent and realistic, replicating production as much as possible and under source control.
4. **Service virtualization** ⚡ Different kinds of testing require different environments. Service virtualizations allow teams to simulate a production environment without the costs and effort associated with creating and managing physical infrastructure.
5. **Testing nonfunctional requirements (NFRs)** ⚡ System attributes, such as security, reliability, performance, scalability, and usability, are critical and require testing.
6. **Continuous integration with suppliers** ⚡ [Suppliers](#) bring unique contributions that reduce lead time and improve value delivery, requiring continuous integration. It helps to adopt a shared integration cadence and establish objective evaluation milestones.

Stage

Finally, the ART must validate the entire solution in staging based on the following practices:

Maintain a staging environment ⚡ A staging environment closely resembling production provides the place for such validation.

Blue/Green deployment ⚡ The blue/green deployment pattern involves two environments—live (production) and idle (staging). Changes flow continuously to idle, stage, and ready them for production deployment. When ready, a configuration change switches the two environments. Idle becomes live, while the old live becomes the new idle. This approach enables continuous delivery, zero-downtime deployment, and fast failure recovery.

System demo ⚡ This is where stakeholders evaluate a solution's readiness for production deployment.

Enabling a Culture of Continuous Integration

Continuously integrating large and complex systems is a time-consuming journey. The following section provides some suggestions for building a thriving CI culture and practice.

Integrate frequently ⚡ The more often teams integrate, the quicker they find

problems. And the harder it is to do, the more often they need to do it. This practice eliminates impediments and adds automation along the way, resulting in faster learning cycles and less rework.

Make integration results visible Ⓛ When the integration process breaks, everybody should know why it failed. When it's fixed, creating new tests should detect the problem earlier and prevent it from happening again.

Fixing failed integrations is a top priority Ⓛ Teams that continue working through integration failures fall short of the values and culture associated with building production-ready systems. Teams often use flashing lights or other notifications, drawing attention to a broken build and establishing visible indicators displaying the percentage of the time the system remains broken.

Establish a shared cadence Ⓛ Integration points are more accessible when all teams follow the same consistent rhythm. Suppose teams cannot do a complete integration within an iteration. In that case, they can make near-term trade-offs on what's possible while continuously improving their techniques and infrastructure toward this goal.

Develop and maintain proper infrastructure Ⓛ Effective continuous integration depends on the availability of test and staging environments. Infrastructure is, of course, an investment. But [Lean-Agile Leaders](#) take the long view and make the investments necessary today to increase velocity for the marathon ahead.

Apply supportive software engineering practices Ⓛ Continuous integration is more accessible when designing systems with those concerns in mind. [Test-first](#) development and planning for testability call for more modular solutions and separation of concerns, as well as using primary interfaces and physical test points.

Another important aspect of CI culture is ensuring a fast flow of value through the pipeline. See the [ART Flow](#) article for more information on making value flow without interruption ([Principle #6](#)).

Enabling Continuous Integration with DevOps

Continuous integration involves crucial "development" activities that originally inspired the "Dev" in DevOps. These activities focus on solution development and pipeline flow through pre-production environments. Applying DevOps thinking, practices, and tooling in this segment of the value stream enables rapid development, frequent code integration, and built-in quality and compliance.

As illustrated in Figure 4, SAFe's CALMR approach to DevOps (center) enables continuous integration and several practice domains (inner rings). Each of the four activities (in green) is a collaborative effort that draws upon DevOps expertise from multiple disciplines to maximize delivery speed and quality.

Figure 4. DevOps enables continuous integration

Figure 4. DevOps enables continuous integration

For example, *building* solutions in the continuous delivery pipeline crosses multiple DevOps domains. Checking code into version control triggers the deployment pipeline to invoke automated merge, quality, and security checks, then apply configurations stored as code to build shippable, full-stack binaries. Using DevOps, this process typically turns source code into tested, deployable solutions quickly.

All four continuous integration activities are enabled by DevOps, though with different combinations of technical practices and tooling. See the DevOps article series for more detailed guidance on DevOps and how it allows the continuous delivery pipeline.

Learn More

[1] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. AMACOM, 2010.

[2] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises (Agile Software Development Series)*. Pearson Education, 2007.

[3] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

Last update: 06 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)

English

Scaled Agile Framework



Specifically, you can take the time to develop and bring to the table an outside-in, market-centric perspective that is so compelling and so well informed that it can counterbalance the inside-out company-centric orientation of last year's operating plan.

—Geoffrey Moore,
Escape Velocity

Adjust Cookie Settings

Continuous Exploration

Continuous Exploration (CE) is an aspect of the Continuous Delivery Pipeline that drives innovation and fosters alignment on what should be built by continually exploring the market and customer needs, defining a vision, roadmap, and set of features for a solution.

Continuous Exploration (CE) is the first aspect of the four-part [Continuous Delivery Pipeline](#) (CDP), which also includes [Continuous Integration \(CI\)](#), [Continuous Deployment](#), and [Release on Demand](#) (Figure 1).

Figure 1. Continuous exploration in the context of the CDP

Figure 1. Continuous exploration in the context of the CDP

During CE, new ideas are raised, refined, and prepared as a list of prioritized

[Features](#) in the [ART Backlog](#). Agile Teams implement the features prioritized by [Product Management](#) during [PI Planning](#), which kicks off the CI process. After that, the CD cycle pulls them into production, where they are validated and prepared for release.

Details

[Agile Product Delivery](#) is one of the seven core competencies of SAFe. It allows the enterprise to deliver increasingly valuable solutions to end users with optimal frequency. CE is integral to that process and focuses on applying [Customer Centricity](#) and [Design Thinking](#) to understand and create alignment on new development opportunities while recognizing that all such ideas are hypotheses that need to be validated.

CE replaces traditional waterfall approaches of up-front, rigid requirement definitions with a process that generates a consistent flow of [Features](#) ready for implementation in the ART Backlog. Decomposing features into small batches of [Stories](#) enables work to move quickly through the remaining aspects of the CDP to the [Customer](#). Getting fast feedback is built into the process, allowing the teams to adjust to market needs. See the [ART Flow](#) article for more information on making value flow without interruption ([Principle #6 - Make value flow without interruptions](#)).

Customers, [Suppliers](#), partners, [Business Owners](#), [Agile Teams](#), [Product Owners](#), and [Lean Portfolio Management](#) are among the internal and external stakeholders involved in this process. Their involvement may be indirect, such as through secondary research on market needs. Or it can be direct, as when Agile Teams participate in an [Innovation and Planning Iteration](#). CE activities enable the organization to align to a shared [Vision](#), a set of features in the backlog defined for implementation, and a forecasted [Roadmap](#).

The Four Activities of Continuous Exploration

Figure 2 illustrates the four steps of continuous exploration, described in the following sections.

Figure 2. Four activities of Continuous Exploration

Figure 2. Four activities of continuous exploration

Hypothesize

Hypothesize describes the practices for generating ideas and the measurements needed to validate them with [Customers](#). Its primary purpose is to define a Solution hypothesis that teams will validate through the CDP.

Product Management has notions of Customer needs based on their understanding of the marketplace, [Strategic Themes](#), [Portfolio Vision](#), and Roadmap. However, these ideas should not be considered facts. Instead, teams should consider them a hypothesis that needs to be tested and proven. Accordingly, the practices associated with hypothesis-driven development include:

Lean startup thinking ▷ Defining *Minimum Marketable Features (MMFs)* and *Minimum Viable Products (MVPs)* [1] helps evaluate hypotheses quickly with minimal investment. MMFs and MVPs represent the smallest amount of usable functionality for early customers, who can provide feedback for future product development.

Innovation accounting ☰ Evaluating hypotheses for a new product or feature requires a different approach than measuring existing solutions. It requires us to consider two questions: 1) Are we progressing toward our outcome hypothesis? 2) How do we know? **Innovation accounting** uses actionable metrics (leading indicators) for determining early results and is a good predictor of future business outcomes. Leading indicators answer these two questions and improve economic decisions during initial solution development and evaluation of the MMF or MVP.

Collaborate and Research

Creating a compelling and differentiated vision requires Product Management to facilitate a continuous and **collaborative** process, soliciting input from diverse stakeholders, as illustrated in Figure 3.

Figure 3. Product Management collaborates with multiple stakeholders to refine requirements

System Architects ☰ **System Architects** have in-depth technical knowledge of solutions. They are responsible for understanding them at the system level and their use cases and **Nonfunctional Requirements (NFRs)**. Although it's natural to view these roles as technically and internally inclined, architects should also have significant and ongoing customer engagement that enables them to identify new ways to solve unmet needs.

Customers ☰ Customers judge value by voting with their wallets or feet. Accordingly, they're the primary source of feedback on the solution and how well it meets their needs. *But a note of caution:* customer motivations are often heavily bound to their current solution context, so they are often motivated only to improve things incrementally. In other words, customer **feedback alone does not constitute a product strategy**. But failing to meet current and evolving customer needs is a sure path to failure.

Business Owners and stakeholders ☰ Business Owners have the business and market knowledge needed to set the mission and vision. A solution that doesn't meet their expectations likely has no value.

POs and teams ☰ Product Owners and Agile Teams create domain expertise through their work creating the solution. In many cases, they are closest to both technical and user concerns. Their input is integral to the ongoing evolution of the solution.

Collaboration and research are grounded in specific practices:

Primary market research ☰ Product Management develops additional insights through primary market research, including surveys, focus groups, questionnaires, and competitive analysis for customer understanding.

Customer visits and Gemba walks ☰ A Gemba walk [2] or customer visit is a process where the product team observes how stakeholders execute the specific activities in their operational value streams to identify opportunities for relentless improvement. There's no substitute for first-person observation of the daily activities of the people doing the work. Whether structured or informal, Product Managers and Product Owners need to understand how people use systems in their work environments. They can't do that at their desk, so there is no substitute for getting outside the building, visiting customers, and observing users in their specific [Solution Context](#).

Secondary market research ☰ To broaden their thinking, Product Management uses various secondary market research techniques to understand the customers and markets they're serving comprehensively. Staying abreast of market/industry trends is a critical outcome of secondary market research.

Lean UX thinking ☰ [Lean UX](#) [5] is a collaborative process of working with stakeholders to define Minimum Marketable Features (MMFs) and validate them quickly with customers.

Collaborative research enables the organization to refine its processes further and create artifacts that clearly express its emerging understanding of the problem space. These include:

Developing personas to focus design ☰ Informed by research, personas help the organization understand their target customers

Building empathy for the user ☰ Empathy maps ensure that the team considers the user's needs and how they may evolve through successive releases

Designing the customer experience ☰ Customer journey maps provide the design link between the operational value stream and the Customer's user experience

While these artifacts tend to be relatively stable over successive releases, the entire enterprise must find ways to avoid making strategic decisions on stale insights.

Architect

With a clear understanding of the problem, CE moves into the solution space, defining the minimum amount of architecture that will support the Solution and enable continuous delivery.

Architects serve the business and the Customer by ensuring the [Architectural Runway](#) is sufficient to deliver the required functionality and is designed to enable the [Continuous Delivery Pipeline \(CDP\)](#). System Architects support the CDP through five practices:

1. **Architecting for releasability** ⚡ Different parts of the solution require distinct release strategies. Therefore, design solutions to enable various incremental release strategies and evolve them over time based on business demand.
2. **Architecting for testability** ⚡ Systems designed and architected modularly enable continuous testing.
3. **Separating deployment and release** ⚡ The capability to deploy continuously requires architectural enablers that allow functionality to be moved into production but hidden from Customers.
4. **Architecting for operations** ⚡ Build telemetry and logging capabilities into every application and solution to meet operational support needs. Allow services to be downgraded or even removed during high loads or in response to incidents. Build capabilities for fast recovery and fix-forward.
5. **Threat modeling** ⚡ Information security considerations should start early, identifying threats to proposed architecture, infrastructure, and applications. Capture essential security requirements as Nonfunctional Requirements to influence backlogs.

Synthesize

Synthesize distills the knowledge gained into a new future state for the solution. The vision, roadmap, and prioritized backlog of features align the ART's teams to a shared direction. Focus synthesis on ensuring these assets are ready for PI planning. The following practices are needed to accomplish this:

Creating the solution vision ⚡ The vision provides the reasons or purpose for developing new features.

Maintaining the solution roadmap ⚡ The ART roadmap provides a view into the near future, helping Product Management prioritize the work, enabling System Architects to prioritize the architecture, and providing visibility for Business Owners.

Defining a backlog with clearly written items ⚡ Defining features that fit in a PI is critical for ARTs to align on what is needed and for teams to plan. The backlog also reflects essential security requirements.

Behavior-driven development (BDD) fosters collaboration between Product

Management, Product Owners, and Agile Teams, which clarifies requirements by adding acceptance criteria.

Economic prioritization ☰ Prioritized features enable effective development. The budget guardrails of capacity allocation, investment horizons, and continuous Business Owners engagement are critical in prioritization.

PI Planning ☰ The exploration work done by the ART is an essential input to the following PI planning event and helps with alignment.

Where there's alignment on what needs to be built, features smoothly flow to the CI segment of the CDP. However, this does not mean that exploration is over. Feedback is continually flowing back from deployed and released features. This feedback informs new decisions about what the ART should work on next and is integral to the CE process.

Enabling Continuous Exploration with DevOps

Activities in continuous exploration set the pace for the entire CDP. Execution is slow when they involve large batches, rigid specifications, and commitments to fixed plans. Thus, for ARTs to achieve continuous delivery, these "upstream" activities should be driven by a bias for speed and validated learning. Applying [DevOps](#) thinking, practices, and tooling early in the value stream reinforces *all* SAFe principles, aligns the entire ART to a DevOps mindset, and primes the CDP.

Many DevOps-related concepts apply at this level. Figure 4 illustrates SAFe's [CALMR](#) approach to DevOps (center) and practice domains (inner rings) support CE. Each of the four activities (in green) is a collaborative effort that draws upon DevOps expertise from multiple disciplines to maximize delivery speed and quality.

Figure 4. DevOps enables continuous exploration

Figure 4. DevOps enables continuous exploration

For example, *architecting* for continuous delivery is not a one-dimensional activity. It crosses several disciplines, as Figure 4 suggests. Agile architecture must account for desired quality and security levels, align to value stream performance objectives, produce tangible configurations under version control, and generate backlog items and NFRs that support Agile planning and emergent design. Moreover, the CALMR mindset should guide all architectural decisions and actions to maximize delivery speed and solution value.

All four CE activities are enabled by DevOps, though with different combinations of technical practices and tooling.

Learn More

[1] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Random House, Inc, 2011.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (Agile Software Development Series)*. Pearson Education, 2011.

[3] Womack, Jim. *Gemba Walks Expanded 2nd Edition*. Lean Enterprise Institute, Inc, 2019.

[4] Gothelf, Jeff, and Josh Seiden. *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media, 2016.

Last update: 06 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



*Our highest priority
is to satisfy the
customer through
early and
continuous delivery
of valuable software.*

~Agile Manifesto

Continuous Delivery Pipeline

The Continuous Delivery Pipeline (CDP) represents the workflows, activities, and automation needed to guide new functionality from ideation to an on-demand release of value.

Figure 1 illustrates the pipeline's four aspects: [Continuous Exploration \(CE\)](#), [Continuous Integration \(CI\)](#), [Continuous Deployment \(CD\)](#), and [Release on Demand](#).

Figure 1. The SAFe Continuous Delivery Pipeline

Figure 1. The SAFe Continuous Delivery Pipeline

The Continuous Delivery Pipeline (CDP) is a significant element of the [Agile Product Delivery](#) competency. Each [Agile Release Train \(ART\)](#) builds and maintains, or shares, a pipeline with the assets and technologies needed to deliver solution value as independently as possible. The first three elements of the CDP (CE, CI, and CD) work together to support the delivery of small batches of new functionality, which are then released to fulfill market demand.

Details

Building and maintaining a CDP allows each ART to deliver new functionality to

users far more frequently than traditional processes. For some, continuous may mean daily or even releasing multiple times per day. For others, it may mean weekly or monthly releases—whatever satisfies market demands and the goals of the enterprise.

Legacy practices often cause ARTs to make solution changes in large monolithic chunks. However, this does not usually need to be an all-or-nothing approach. For example, a satellite system comprises a manufactured orbital object, a terrestrial station, and a web farm that feeds the acquired data to end users. Some components may be released daily—perhaps the web farm functionality or satellite software. Other elements, like the hardware components, can only be done once every launch cycle.

Decoupling the web farm functionality from the physical launch eliminates the need for a monolithic release. It also increases Business Agility by allowing the delivery of solution components in response to frequent market changes.

The Four Aspects of the Continuous Delivery Pipeline

The SAFe CDP contains four aspects: continuous exploration, continuous integration, continuous deployment, and release on demand. The CDP enables organizations to map their current pipeline into a new structure and use relentless improvement to deliver value to customers. Internal feedback loops often identify process improvements, while external feedback identifies solution improvements. The improvements collectively create synergy in ensuring the enterprise is “building the right thing, the right way” and frequently delivering value to the market. The paragraphs below describe each aspect.

Continuous Exploration (CE) focuses on creating alignment on what needs to be built. In CE, design thinking ensures the enterprise understands the market problem or customer need and the solution required to meet that need. It starts with a hypothesis of something that will provide value to customers. Ideas are then analyzed and further researched, leading to the understanding and convergence of the requirements for a Minimum Viable Product (MVP) or Minimum Marketable Feature (MMF). These feed the solution space for exploring how existing architectures and solutions can or should be modified. Finally, convergence occurs by understanding which **Capabilities** and **Features**, if implemented, are likely to meet customer and market needs. Collectively, these are defined and prioritized in the ART Backlog.

Continuous Integration (CI) focuses on taking features from the ART backlog and implementing them. In CI, the application of design thinking tools in the problem

space focuses on the refinement of features (for example, designing a user story map), which may motivate more research and the use of solution space tools (such as user feedback on a paper prototype). After specific features are clearly understood, Agile Teams implement them. Completed work is committed to version control, built and integrated, and tested end-to-end before being validated in a staging environment.

Continuous Deployment (CD) takes the changes from the staging environment and deploys them to production. At that point, they're verified and monitored to ensure they are working correctly. This step makes the features available in production, where the business determines the appropriate time to release them to customers. This aspect allows the organization to respond, rollback, or fix forward when necessary.

Release on Demand (RoD) is the ability to make value available to customers together or in a staggered fashion-based on market and business needs. This approach allows the business to release when market timing is optimal and carefully controls the risk associated with each release. Release on demand also encompasses critical pipeline activities that preserve solutions' stability and enduring value long after release.

Although described sequentially, the pipeline isn't strictly linear. Instead, it's a learning cycle that allows teams to establish one or more hypotheses, build a solution to test each one, and learn from that work, as Figure 2 illustrates.

Figure 2. The CDP fosters continuous learning and value delivery

Figure 2. The CDP fosters continuous learning and value delivery

Although a single feature flows through the Value Stream sequentially, the teams work through all aspects in parallel. That means that ARTs and Solution Trains, throughout every PI and every iteration in the PI, continuously:

- Explore user value
- Integrate and demo value
- Continuously deploy to production
- Release value whenever the business needs it

Start by Mapping the Current Workflow

Successful enterprises already have a delivery pipeline—otherwise, they wouldn't be able to release any value. But too often, they are not fully automated, contain significant delays, and require tedious and

Value Stream Mapping

error-prone manual intervention. These factors cause organizations to delay releases, increasing their size and scope (ØWeÕll release when it is big enough).

This approach is the opposite of the [SAFe Principle #6](#), which makes value flow without interruption.

The first step to improving value flow is mapping the current pipeline. Figure 3 illustrates the flow of value through one enterpriseÕs CDP, focusing initially on new Feature development. Over time, this map would be extended to capture any change to the system, from new Features to maintenance to architectural improvements.

Figure 3. A map of one company's delivery pipeline

Figure 3. A map of one companyÕs delivery pipeline

Once the current pipeline has been mapped, metrics can be collected and recorded on the value stream map to understand where delays occur. These metrics enable the ART to identify opportunities for improvement (such as eliminating delays or reducing rework). Four primary metrics [1] are used (Figure 4):

Process time is the time it takes to get work done in one step. For example, in Figure 4, the "Design" step takes four hours.

Lead time is the time it takes from when the work was done in the previous step until it's done in the current step. In other words, *lead time = delay time from the last step + process time* of the current step. In Figure 4, the lead time from creating ideas to defining them is variable. It is common when first mapping systems not to have metrics on specific steps. In this case, the ART can improve the remaining process while metrics are gathered on the variable part.

Delay time is the time when no work is happening. Figure 4 shows that the features accepted by the Product Manager are delayed a staggering 696 hours before being deployed to staging. Understanding and eliminating unnecessary delays is critical to improving the flow of value.

Percent complete and accurate (%C&A) represents the percentage of work that the next step can process without needing rework. Often, delays are caused by poor quality in the upstream (prior) steps. The %C&A metric helps identify the steps where poor quality might be occurring and causing longer lead times, resulting in delays in value delivery. Figure 4 indicates that 20% of the time, the work moving from the "Design" step to the "Code" step needs to be reworked. Improving the %C&A metric is also essential to improving the flow of value. The %C&A of a single step is extended to *rolled %C&A*, a measure that captures the likelihood that an item will pass through the *entire* workflow without rework. With a cumulative rolled %C&A of 35%, this workflow is reworking more than half of its steps.

Figure 4. Value stream map with flow metrics

Figure 4. Value stream map with flow metrics

Align the Current Workflow to the Continuous Delivery Pipeline

Once the current flow is understood, it can be mapped into the SAFe CDP. Mapping helps the organization adopt a shared mental model and efficiently communicate changes and improvements. Figure 5 removes the continuous labels because, at this stage, the process is unlikely to resemble an automated pipeline.

Figure 5. SAFe's Continuous Delivery Pipeline mapped to a value stream

Figure 5. SAFe's Continuous Delivery Pipeline mapped to a value stream

Identify Opportunities for Improvement

Teams look for the opportunity to improve the efficiency of each step, consequently reducing the total lead time. This improvement includes addressing process time and each step's quality (percent complete and accurate). The higher that number, the less rework is required and the faster the work moves through the system.

As shown in Figure 6, the delay time between steps is often the most significant initial factor. Delay time represents handoffs, waiting, and other non-value-added wastes. This process has two considerable delays and a substantial amount of rework in the first step of the deployment process. Reducing delays is typically the fastest and easiest way to lower the total lead time. Another high-priority area to improve is any step with low %C&A metrics, as reducing rework enables the ART to focus on creating value (for example, for a software solution, instead of fixing bugs, the team can focus on new features). Subsequent opportunities for improvement

focus on reducing the batch size and applying the DevOps practices identified in each of the specific articles describing the continuous delivery pipeline. The [Team Flow](#) and [ART Flow](#) articles provide further guidance on how to make value flow without interruption (Principle #6).

Figure 6. Value stream maps reveal major delivery bottlenecks

Figure 6. Value stream maps reveal major delivery bottlenecks

Tracking Continuous Delivery

Continuous delivery is an extensive process. Indeed, it may be the most vital capability of every ART and Solution Train. [Product Management](#) and its stakeholders should visualize and track ongoing work, even though a significant portion of the development is automated. The ART Kanban facilitates the flow of [Features](#) through the CDP. Work in Process (WIP) limits help improve throughput and identify and address bottlenecks. Figure 7 illustrates a typical ART Kanban, example policies, and WIP limits governing each state.

Figure 7. An example ART Kanban board

Figure 7. An example ART Kanban board

The Kanban systems consist of a series of states, each of which is summarized below:

Funnel ⚡ This is the capture state for all new features or enhancement of existing system features.

Analyzing ⚡ Features that best align with the vision are pulled into the analyzing step for further exploration. Here they're refined with critical attributes, including the business benefit hypothesis and acceptance criteria.

Ready ⚡ After analysis, higher-priority features move to the backlog, where they're ranked.

Implementing ⚡ At every PI boundary, top features from the ART backlog are pulled into the implementing stage, where they're developed and integrated into the system baseline.

Validating on staging ⚡ Features ready for feedback get pulled into this step to be integrated with the rest of the system in a staging environment and then tested and

validated.

Deploying to production ⚡ When capacity is available, features are deployed into the production environment, where they await release.

Releasing ⚡ Features are released once a sufficient amount of value has been created to meet market demands and the benefit hypothesis is evaluated.

Done ⚡ When the hypothesis has been satisfied, no further work on the feature is necessary, and it moves to the done column.

Enable the Continuous Delivery Pipeline with DevOps

Building, maintaining, and optimizing a continuous delivery pipeline requires specialized skills and tools throughout the entire value stream. Because this type of delivery system calls for the rapid delivery of complex solutions with very short learning loops and high degrees of cross-functional collaboration, DevOps methods are ideally suited to enable it. In other words, continuous delivery pipelines are best implemented with DevOps, as illustrated in Figure 8.

Figure 8. DevOps enables the CDP

Figure 8. DevOps enables the CDP

SAFe's [CALMR](#) approach to DevOps is a mindset that guides continuous value delivery by improving *Culture, Automation, Lean Flow, Measurement, and Recovery*. DevOps technical skills, practices, and tooling are grouped into *practice domains*, represented by the model's inner loops. The two outer loops represent the four aspects of the CDP, each of which has four activities.

Download the free DevOps Health Radar assessment [here](#). AgilityHealth also offers an [online version of this assessment](#).

Learn More

- [1] Martin, Karen. *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation*. McGraw-Hill Education, 2013.
- [2] Kim, Gene. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*. IT Revolution Press, 2013.
- [3] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

Last update: 5 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

TRAINING

BUSINESS

[YouTube](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



While we must acknowledge emergence in design and system development, a little planning can avoid much waste.

—James Coplien,
Lean Architecture

Architectural Runway

[Adjust Cookie Settings](#)

The Architectural Runway consists of the existing code, components, and technical infrastructure needed to implement near-term features with minimal redesign and delay.

Architectural Runway enables a continuous flow of value through the [Continuous Delivery Pipeline](#), providing the technology required to quickly define, build, validate, and release [Features and Capabilities](#). It also supports the practice of [Agile Architecture](#) by allowing an organization's technology landscape to evolve in response to changing business needs.

Details

Agile development avoids big design up-front (BDUF) with a simple belief that the best architectures, requirements, and designs emerge from self-organizing teams [1]. This yields the practice of *emergent design*—defining and extending the architecture only as necessary to deliver the next increment of functionality.

However, emergent design alone cannot handle the complexity of large-scale [Solution](#) development. At scale, relying solely on emergent design leads to the following problems:

- Lack of standards increases delivery costs and delays
- One-off solutions become difficult to change and maintain
- Systems become vulnerable to security and stability issues
- Quality becomes dependent on tribal knowledge
- Solution components have poor interoperability and reusability

These problems can result in poor solution performance, unfavorable economic outcomes, and delayed time-to-market. Organizations overcome these problems by balancing emergent design with *intentional architecture*, which requires some centralized planning and cross-team coordination. Both are implemented with [Enablers](#) and, together, pave the architectural runway (Figure 1).

Figure 1. Architectural runway evolves in support of dynamic business needs

Figure 1. Architectural runway evolves in support of dynamic business needs

Intentional Architecture Supports System Evolution

Agile Teams apply emergent design through the development of enabler Stories that incrementally evolve their parts of the solution. However, they typically lack the breadth of knowledge of the end-to-end solution that would allow them to design components and infrastructure across all domains effectively. Many of these design factors are simply beyond their scope of responsibility.

Intentional architecture accounts for these systemic design considerations and provides purposeful, planned architectural guidelines that enable the independent work of Agile teams to be integrated into a cohesive, sustainable solution. These cross-cutting guidelines are typically defined by Enterprise Architects, Solution Architects, and System Architects in close collaboration with Product Management and

[Solution Management](#). These roles are well suited to the task because of their broad knowledge of the technology landscape and deep knowledge of the [Solution Context](#).

Intentional architecture is codified as enabler epics, capabilities, and features in the Portfolio Backlog, Solution Backlog, and ART Backlog. Architects then help steer the enablers through the appropriate Kanban systems, ensuring they produce the intended architectural runway.

Building the Architectural Runway

Several roles may be involved in defining architectural runway, but its implementation is the responsibility of Agile teams. These teams are often responsible for delivering end-user-focused products or services. Therefore, building the architectural runway should not overly constrain them.

Just the right amount of architectural runway is required at any given time. Too much, and the architecture bottlenecks the teams and over-engineers the current solution increment. Too little, and the organization will not have the runway it needs to meet near-term business commitments. Capacity allocation is applied to the [Team Backlog](#) to help ensure that the ratio of enabler work to customer-facing work is always in balance.

A dedicated Agile team often oversees the initial implementation when significant investment in architectural runway is called for—such as to enable a new product launch, Horizon 3 portfolio initiative, or legacy environment. (Figure 2).

Figure 2. A dedicated Agile team establishing initial architectural runway

Figure 2. A dedicated Agile team establishing initial architectural runway

In this scenario, an architect often assumes the role of [Product Owner](#), acting as the voice of the customer and content authority over a backlog composed primarily of enablers. A specialized group of team members handles development, and a Scrum Master/Team Coach guides execution.

This team continues until the volume of runway work required by the ART is no longer sufficient to warrant a dedicated team. At this point, the responsibility for implementing additional architectural runway as needed becomes shared among the ART's persistent Agile teams, as illustrated in Figure 3.

Figure 3. Agile teams share responsibility for extending the architectural runway

Figure 3. Agile teams share responsibility for extending the architectural runway

Regardless of who performs the work, the rules for building the runway are both simple and Agile:

- Teams that build the runway iterate like every other Agile team on the ART
- The value is in working systems, not models or specifications
- Enabler features and stories should take no more than a PI or iteration, respectively, to deliver
- Agile teams are stakeholders of the architectural runway, leveraging it to deliver on customer commitments and providing feedback on its effectiveness

Continually Invest in the Architectural Runway

Architectural runway is dynamic. It is consumed by delivering near-term features and must be extended to support future features. The following are examples of forces that consume architectural runway:

Fast-moving Agile teams ⚡ They quickly use the newest runway to deliver near-term features

Customer preferences ⚡ After investing in architectural runway, stakeholders often shift backlog priorities to the features that directly benefit customers

Technology innovation ⚡ Technology changes rapidly, rendering existing runway obsolete

Changing business needs ⚡ The needs of the enterprise shift in response to emerging opportunities and threats

Architectural runway is consumed to deliver planned features. As business needs change and new features are requested, additional runway is required (Figure 4).

Figure 4. Architectural runway is produced and consumed as business needs evolve

Because the development of new features and capabilities consumes the architectural runway, a continual investment must be made to extend it. Teams commit to extending the runway as needed in each iteration to support quick, sustainable delivery velocity. This could include adding automation to the CDP, enhancing [DevOps](#) practices, increasing server capacity, or any other activity that

accelerates delivery velocity.

The Architectural Runway in Large Solutions

When building really big systems, the architectural runway takes on an even more critical role as multiple ARTs contribute to the same Solution as part of a Solution Train. The [Enterprise Solution Delivery](#) article describes ten Lean-Agile practices that guide large solution delivery, several of which relate directly to architectural runway:

Specify the solution incrementally ☰ The [Solution Intent](#) defines many constraints as Non-Functional Requirements (NFRs). Many NFRs are cross-cutting concerns that can be addressed and simplified by building architectural runway that supports integration and testing. Architectural runway also allows elements of the solution intent to smoothly and incrementally evolve from variable to fixed.

Apply multiple planning horizons ☰ Large solutions generally require a longer architectural runway, implemented with enabler epics over several PIs or even years. Efficient delivery of these long stretches of runway is orchestrated through connected iteration plans, PI roadmaps, and solution roadmaps.

Design for change ☰ Achieving these goals in large solutions requires intentional architecture to implement an effective Continuous Delivery Pipeline and ensures simplified operations with strong feedback via application telemetry.

Continually address compliance concerns ☰ A Lean approach to [Compliance](#) automates the collection and testing of compliance data to provide more effective and predictable outcomes. This goal requires early engagement with auditors and stakeholders to agree on an acceptable approach. Creating capabilities through the architectural runway ensures consistency and removes significant compliance risks.

Evolve deployed systems ☰ A fast, economical Continuous Delivery Pipeline means solutions and changes to those solutions can be released on demand. Architectural runway delivers a CDP whereby deployed solutions can evolve incrementally with minimal customer disruption.

It is critical to implement long runway incrementally. Enabler epics are split into enabler features and/or capabilities, then implemented by ARTs. Each enabler feature must be completed within a PI, and enabler stories within an iteration. This allows significant investments in architectural runway to evolve with the right balance of intentional architecture and emergent design.

The Backstory of the Architectural Runway

The term “architectural runway” started as an analogy while observing PI-level burn-down charts. Often, when there isn’t enough architectural runway when

teams start a PI, any features dependent on the new architecture are high risk.

ARTs can't always land those PIs (bring the burn-down to zero at the end of the PI). In that case, they don't meet the PI objectives. A PI, like an airplane, needs enough runway to land safely.

In SAFe, the architectural runway line goes up and down over time because it is built up, then used, then built up some more, then used, and so on. There has to be just the right amount at any point to support near-term business objectives.

To extend the runway metaphor, the bigger the aircraft (system) and the faster the flying speed (velocity), the more runway is needed to land the PI safely.

Learn More

[1] Manifesto for Agile Software Development. <http://www.agilemanifesto.org>

Last update: 9 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300

[Blog](#)

[Twitter](#)

[Usage and Permissions](#)

Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Linkedin](#)

[YouTube](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



[A] system must be continually adapted, or it becomes progressively less satisfactory.

~ Manny Lehman

System Architect

[Adjust Cookie Settings](#)

The System Architect is responsible

Find a Course:

[Go](#)

for defining and communicating a shared technical and architectural vision for the solutions developed by an ART.

This helps ensure that the system or solution under development fits its intended purpose.

Details

System Architects play a critical role in aligning [Agile Release Train](#) teams to a shared technical direction. They partner with their teams to:

- Elaborate the system architecture
- Validate technology assumptions
- Evaluate implementation alternatives
- Create the continuous delivery pipeline

In ARTs not part of a [Solution Train](#), System Architects also perform many of the responsibilities of [Solution Architects](#).

To navigate the complexities of system architecture productively, an ART may have multiple System Architects steering different aspects of the system architecture, working together toward a common business goal.

Key Collaborations

System architecture significantly impacts the solution, the teams that develop it, and the broader enterprise ecosystem. Consequently, a critical responsibility of the System Architect is collaborating with various roles, teams, and stakeholders within and outside the ART. The primary goal of these interactions is to ensure that the ART advances a system architecture that supports the evolving business need and enables fast reliable implementation. The key collaborations of the System Architect are shown in Figure 1.

Figure 1. Key collaborations of the System Architect

Figure 1. Key collaborations of the System Architect

The most critical interactions appear in the following areas:

Steer the ART ☰ The System Architect works with [Product Management](#) and the [Release Train Engineer](#) (RTE) to provide the necessary guidance to the train. In this association, Product Management helps define business priorities, the architect ensures that the architecture supports the need, and the RTE provides the necessary facilitation of the train's core processes. Additionally, the System Architect, as well as the RTE and the Product Management, work with the ART's [Business Owners](#) to ensure alignment with the organization's higher strategic intent. Many of these interactions occur around PI boundaries in preparation for and during [PI Planning](#). Besides steering the solution development effort, the three roles lead [Value Stream Management](#) activities for the ART.

Align on solution/enterprise architecture ☰ The System Architect works proactively with the [Enterprise Architect](#) to ensure that the ART is aligned with the organization's technology landscape. However, when the train is working on a larger solution that involves other ARTs and suppliers, the System Architect interacts with the Solution Architect (as well as other System Architects from the Solution Train).

This work includes defining and supporting the implementation of enabler capabilities, consistent choice and use of technologies, regulatory compliance, standards, and higher-level NFRs.

Evolve system architecture ☰ The System Architect routinely interacts with [Agile Teams](#) (including [Product Owners](#) and [Scrum Master/Team Coaches](#)) to ensure that system design evolves in line with a shared architectural intent and the practical reality of implementation. Some of these interactions happen at PI Planning, where the architect ensures teams' alignment with the architectural vision. However, the collaboration continues throughout the PI execution. Here, the architect and teams work together to learn and adjust the course of action based on new facts that arose from implementation.

Work with other groups ☰ The System Architect addresses other concerns with groups like the [System Team](#) and [Shared Services](#). The goal of interacting with the System Team is usually to establish optimal architectures and processes for system integration and testing. Often the System Team is significantly involved in implementing critical architectural enablers, which usually results in a close working relationship with the System Architect. Depending on the context, interactions with Shared Services may involve having the System Architect aid the design of the deployment pipeline, as well as information architecture, regulatory compliance issues, information security, metrics instrumentation, and so on.

Responsibilities

The System Architect's responsibilities consist of the five areas shown in Figure 2.

Figure 2. Areas of responsibility of a System Architect

Figure 2. Areas of responsibility of a System Architect

This article further describes each of the areas.

Aligning Architecture with Business Priorities

The purpose of every solution architecture is to support the development and delivery of business value. Aligning with the business intent is a multi-faceted area of the System Architect's responsibilities, which include:

Define enablers and architectural runway ▷ The System Architect will define the necessary enablers, and the ART will progressively build up the runway to support the intended features. The architect's role is to continuously define, adjust, and support the journey. Defining new enabler features and advancing the runway happens progressively, with critical checkpoints at every PI boundary.

Participate in solution definition ▷ As a part of [Design Thinking](#), the architect is often closely involved in defining the solution. For solution ideation to be effective, it

must occur in close connection with the reality of the technological and implementation capabilities of the train. The architect provides critical insights in this regard.

Define system NFRs ☐ The System Architect defines [Nonfunctional Requirements](#) (NFRs) for the solution and ensures that the architecture will support the required NFRs. System Architects also assist the train in determining specific measures and the necessary instrumentation to safeguard and monitor NFRs.

Ensure capacity allocation for enablement work ☐ Advancing architectural capabilities requires the teams' time and effort. The System Architect works with Product Management to allocate proper capacity for architecture work, which occurs while preparing for each PI boundary and during the PI Planning itself.

Defining and Communicating Architecture Vision

The complexity of today's technology solutions can be overwhelming; to reduce its impact on teams and maintain productivity, the architect embraces a vital role of communicating a clear technology intent to the teams and the rest of the ART. This creates alignment with the system design and the implementation strategy, ultimately resulting in a faster, more sustainable flow of customer value. In achieving this goal, the System Architect will do the following:

Present architectural vision to teams during PI planning ☐ The System Architect updates the architectural vision before every PI boundary. During the PI planning, the architect presents the vision as a part of the briefings and stays available to the teams during the rest of the planning activities. The architectural vision is repeatedly updated to accommodate new and essential technical aspects relevant to the upcoming PI.

Provide guidance on implementing the vision ☐ During the PI execution, the System Architect stays in sync with the teams and provides further guidance as needed. It's common for the initial architecture vision to change due to new insights that occur during implementation. The architect ensures that all the system design issues teams encounter during the implementation are addressed.

Architect for agility and change ☐ To respond effectively to change and embrace new facts that emerge during development, the system design must enable flexibility (see [Agile Architecture](#) for additional detail). Helpful tools that allow preserving options include balanced, intentional use of abstraction in system design and [Set-Based Engineering](#).

Evolving System Design with Teams

Any architecture requires ongoing adjustment. So, a critical role of the architect is to establish sustainable

interactions with teams that will help reveal inconsistencies in the architectural intent and assist the ART in making necessary improvements. This is a continuous process, a loop of adjusting the intent and learning from the implementation. In this regard, system Architects focus on the following areas:

Support architectural experiments and spikes ☰ The System Architect works with teams to define and realize crucial experiments to validate architectural ideas with the minimum development effort. Because untested architectural assumptions can be extremely costly when they turn out to be incorrect, this dramatically reduces the amount of rework.

Collaborate with teams on optimal system design ☰ The architect frequently interacts with teams to provide new inputs and get feedback on the current architectural intent. This collaboration often occurs as part of PI planning, [Systems Demos](#), and [Inspect and Adapt](#). Additionally, the architect may participate in individual team events whenever significant architectural developments are needed. The System Architect provides ongoing coaching to help teams acquire a bigger-picture view of the system design and increase their expertise as they advance the runway to support new business features. To promote a shared understanding of the architectural approach and the technologies involved in solution development, it's common for a System Architect to participate in technical [Communities of Practice](#) (CoPs).

Align architectural intent with the reality of implementation ☰ The architect makes sure that there is no substantial gap between the expectations expressed in the architectural vision and the reality of team capacities, skills, tools, etc. For this alignment to occur and continue, the architect must stay in close contact with both the teams and the solution assets. This is often achieved by having the System Architect participate directly in creating solution assets or peer reviews. Additionally, it's helpful for the architect to be directly involved in architectural research spikes that teams undertake.

Fostering Built-in Quality and Attending to NFRs

In many ways, architecture determines how easy or hard it will be to implement new solution features that support quality standards for the new functionality without breaking the old. Likewise, architecture significantly impacts the ART's ability to implement and sustain the solution's NFRs. To support these areas of concern, System Architects do the following:

Promote system design that supports Built-in Quality ☰ the System Architect defines architectures that help the ART shift learning left and discover problems early

or prevent them in the first place. Loosely coupled architectures diligently followed design and coding standards, and ongoing refactoring help keep the system healthy. This results in improved testability and maintainability of code and other solution assets.

Attend to the system NFRs ☰ NFRs influence solution viability and sustainability significantly. If not adequately attended to throughout development, they may create a large volume of rework or even thwart the entire solution development effort. The System Architect provides ongoing guidance to the train to help incrementally build and sustain NFRs. The architect often assists teams in devising an effective implementation strategy for new NFRs and helps create standards for maintaining the existing ones (some of which are reflected in the corresponding Definition of Done).

Supporting DevOps and the Continuous Delivery Pipeline

System architecture is intricately connected to customer value delivery. This is because architectures can either make value delivery a fast, incremental process the team and stakeholders can control or a nightmare with unpredictable timelines and harsh consequences for the customer and the business. Additionally, the [Continuous Delivery Pipeline](#) (CDP) has a structure that needs to be defined in a manner that best supports the needs of the ART.

Participate in the release governance process ☰ The System Architect helps the ART develop system architecture in a way that supports incremental value delivery. Additionally, the System Architect provides valuable input and assesses the technology impact for a specific release.

Support design of the CDP ☰ System Architects promote ideas for continuous delivery and DevOps. They also help the teams define the specific architecture of the deployment pipeline, the environments, their procurement, and the necessary tools involved. As a part of this effort, the System Architect helps the ART determine suitable infrastructure configurations. This often includes using [Cloud](#) architectures to enhance the scalability and flexibility of the CDP.

Enable metrics instrumentation ☰ The System Architect facilitates system design that supports the implementation of all necessary metrics. This involves special platforms and tools suitable for the task and proper integration with the solution. In addition, System Architects often accept responsibility for supporting experimentation and measurement of user behavior in [LeanUX](#) by designing working prototypes or creating specific capabilities in the solution itself.

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



What if we found ourselves building something that nobody wanted? In that case, what did it matter if we did it on time and on budget?

—Eric Ries

Lean User Experience (Lean UX) is a team-based approach to building better products by focusing less on the theoretically ideal design and more on iterative learning, overall user experience, and customer outcomes.

Lean UX design extends the traditional UX role beyond merely executing design elements and anticipating how users might interact with a system. Instead, it encourages a far more comprehensive view of why a [Feature](#) exists, the functionality required to implement it, and the benefits it delivers. By getting immediate feedback to understand if the system will meet the fundamental business objectives, Lean UX provides a closed-loop method for defining and measuring value.

Details

Generally, UX represents a user's perceptions of a system—ease of use, utility, and the user interface's (UI) effectiveness. UX design focuses on building systems that demonstrate a deep understanding of end users. It considers users' needs and wants while making allowances for their context and limitations.

When using Agile methods, a common problem is how best to incorporate UX design into a rapid [Iteration](#) cycle, resulting in a full-stack implementation of the new functionality. When teams attempt to resolve complex and seemingly subjective user interactions while simultaneously trying to develop incremental deliverables, they can often churn through many designs, creating frustration with Agile.

Fortunately, the Lean UX movement addresses this using Agile development with Lean Startup implementation approaches. The mindset, principles, and practices of SAFe reflect this thinking. This process often begins with the SAFe Lean Startup Cycle described in the [Epic](#) article. It continues developing [Features and Capabilities](#) using the Lean UX process described here.

As a result, [Agile Teams](#) and [Agile Release Trains \(ARTs\)](#) can leverage a common strategy to generate rapid development, fast feedback, and a holistic user experience that delights users.

The Lean UX Process

In *Lean UX*, Gothelf and Seiden [2] describe a model we have adapted to SAFe, as Figure 1 illustrates.

Figure 1. The Lean UX Process (adapted from Ref [2])

Figure 1. The Lean UX Process (adapted from Ref [2])

Benefit Hypothesis

The Lean UX approach starts with a benefit hypothesis: Agile teams and UX designers accept that the right answer is unknowable up-front. Instead, teams apply Agile methods to avoid Big Design Up-front (BDUF), focusing on creating a hypothesis about the feature's expected business result. Then they implement and test that hypothesis incrementally.

The SAFe Feature and Benefits matrix (FAB) can be used to describe the hypothesis as it moves through the [Continuous Exploration](#) aspect of the CDP:

Feature ▷ A short phrase giving a name and context

Benefit hypothesis ▷ The proposed measurable benefit to the end-user or business

Note: Design Thinking practices suggest changing the order of the feature benefit hypothesis elements to identify the customer benefits first and then determine what features might satisfy their needs.

Outcomes are measured in the [Release on Demand](#) aspect of the CDP. They are best done using leading indicators (see [Innovation Accounting](#) in [1]) to evaluate how well the new feature meets its benefits hypothesis. For example, “We believe the administrator can add a new user in half the time it took before.”

Collaborative Design

Traditionally, UX design has been an area of high specialization. People with a talent for design, a feel for user interaction, and specialty training are often entirely in charge of the design process. The goal was “pixel perfect” early designs, done before the implementation. But this work was often done in silos by specialists that may or may not know the most about the system and its context. Success was measured by how well the implemented user interface complied with the initial UX design. In Lean UX, this changes dramatically:

“Lean UX has no time for heroes. The entire concept of design as a hypothesis immediately dethrones notions of heroism; as a designer, you must expect that many of your ideas will fail in testing. Heroes don’t admit failure. But Lean UX designers embrace it as part of the process.” [2]

Continuous exploration takes the hypothesis and facilitates an ongoing and collaborative process that solicits input from a diverse group of stakeholders — [Architects](#), [Customers](#), [Business Owners](#), [Product Owners](#), and [Agile Teams](#). This group further refines the problem and creates artifacts that clearly express the emerging understanding, including personas, empathy maps, and customer experience maps (see [Design Thinking](#)).

Agile teams are empowered to design and implement collaborative UX, significantly improving business outcomes and time-to-market. Moreover, another important goal is to deliver a consistent user experience across various system elements or channels (for example, mobile, web, kiosk) or even different products from the same company. Enabling this consistency requires balancing decentralized control with centralizing certain reusable design assets (following [Principle #9 – Decentralize decision-making](#)). For example, creating a design system [2] with a set of standards that contains whatever UI elements ARTs and [Value Streams](#) find helpful, including:

Editorial rules, style guides, voice and tone guidelines, naming conventions, standard terms, and abbreviations

- Branding and corporate identity kits, color palettes, usage guidelines for copyrights, logos, trademarks, and other attributions
- UI asset libraries, which include icons and other images, templates, standard layouts, and grids
- UI widgets, which include the design of buttons and other similar elements

These centralized assets are integral to the [Architectural Runway](#), which supports decentralized control while recognizing that some design elements must be centralized. After all, these decisions are *infrequent, long-lasting*, and provide *significant economies of scale* across both the user base and enterprise applications, as described in [Principle #9](#).

Building a Minimum Marketable Feature

With a hypothesis and design, teams can implement the functionality as a Minimal Marketable Feature (MMF). The MMF should be the smallest amount of functionality that must be provided for a customer to recognize any value and for the teams to learn whether the benefit hypothesis is valid.

By creating an MMF, the ARTs apply SAFe [Principle #4](#) *D Build incrementally with a fast, integrated learning* cycle to implement and evaluate the feature. Teams may preserve options with [Set-Based Design](#) as they define the initial MMF.

In many cases, extremely lightweight and not even functional designs can help validate user requirements (ex., paper prototypes, low-fidelity mockups, simulations, API stubs). In other cases, a vertical thread (full stack) of just a portion of an MMF may be necessary to test the architecture and get fast feedback at a [System Demo](#). However, in some instances, the functionality may need to proceed to [deployment](#) and release, where application instrumentation and telemetry provide feedback data from production users.

Evaluating

MMFs are evaluated as part of deploying and releasing (where necessary). There are various ways to determine if the feature delivers the proper outcomes. These include:

- Observation** *D* Wherever possible, directly observe the actual usage of the system. It's an opportunity to understand the user's context and behaviors.
- User surveys** *D* A simple end-user questionnaire can obtain fast feedback when direct observation isn't possible.
- Usage analytics** *D* Lean-Agile teams build analytics into their applications, which

helps validate initial use and provides the application telemetry needed to support a [Continuous Delivery](#) model. Application telemetry offers constant operational and user feedback from the deployed system.

A/B testing This is a form of statistical hypothesis comparing two samples, which acknowledges that user preferences are unknowable in advance. Recognizing this is liberating, eliminating endless arguments between designers and developers—who likely won't use the system. Teams follow [Principle #3 – Assume variability; preserve options](#) to keep design options open as long as possible. And wherever it's practical and economically feasible, they should implement multiple alternatives for critical user activities. Then they can test those other options with mockups, prototypes, or even full-stack implementations. In this latter case, differing versions may be deployed to multiple subsets of users, perhaps sequenced over time and measured via analytics.

In short, measurable results deliver the knowledge teams need to refactor, adjust, redesign—or even pivot to abandon a feature based solely on objective data and user feedback. Measurement creates a closed-loop Lean UX process that iterates toward a successful outcome, driven by evidence of whether a feature fulfills the hypothesis.

Implementing Lean UX in SAFe

Lean UX differs from the traditional, centralized approach to user experience design. The primary difference is how the hypothesis-driven aspects are evaluated by implementing the code, instrumenting where applicable, and gaining user feedback in a staging or production environment. Implementing new designs is primarily the responsibility of the Agile Teams, working in conjunction with Lean UX experts.

Of course, this shift, like many others with Lean-Agile development, can cause significant changes to the way teams and functions are organized, enabling a continuous flow of value. For more on coordinating and implementing Lean UX—specifically how to integrate Lean UX in the PI cycle—read the advanced topic article [Lean UX and the PI Lifecycle](#).

Learn More

[1] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Random House, Inc, 2011.

[2] Gothelf, Jeff, and Josh Seiden. *Lean UX: Designing Great Products with Agile*

Teams. O'Reilly Media, 2016.

[3] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 21 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Imagine a world where product owners, Development, QA, IT Operations, and Infosec work together, not only to help each other, but also to ensure that the overall organization succeeds. By working toward a common goal, they enable the fast flow of planned work into production, while

Adjust Cookie Settings

achieving world-class stability, reliability, availability, and security. [1]

~The DevOps Handbook

DevOps

Note: This is the **home page** for the three-part **SAFe DevOps** series. This article is the **first** in the series which introduces the foundational DevOps concepts. The following links provide access to the next articles in the series: [A CALMR Approach to DevOps](#) and [SAFe's DevOps Practice Domains](#).

DevOps is a mindset, culture, and set of technical practices that supports the integration, automation, and collaboration needed to effectively develop and operate a solution.

DevOps is part of the [Agile Product Delivery](#) competency and is a combination of two words: *development* and *operations*. Without DevOps, there is often significant tension between those who build [Solutions](#) and those who support and maintain them. DevOps helps break down organizational silos and develop a [Continuous Delivery Pipeline \(CDP\)](#)—a high-performance innovation engine capable of delivering market-leading solutions at the speed of business.

Figure 1. The Continuous Delivery Pipeline

Figure 1. The Continuous Delivery Pipeline

The goal of DevOps is simple, to deliver value whenever there is a business need. Indeed, teams that successfully adopt DevOps, on average: deploy 208 times more frequently, 106 times faster, experience seven times fewer failures, and recover from incidents 2,604 times faster than low-performing teams. [2]

DevSecOps

DevSecOps is a term that emphasizes the importance of proper information **security** practices in the pursuit of continuous delivery. Because the origins of DevOps did not explicitly include security as a top-level concern (as it did for development and operations), DevSecOps has emerged as a popular term that avoids any risk of security being an afterthought.

The security community has evolved DevOps thinking beyond its development and operations roots. The **State of DevOps Report**—the world’s longest-running and most widely cited DevOps research project—has revealed that an organization’s

security improves when it's wholly integrated into the [Value Stream](#). [3] In one of the most-read DevSecOps articles on the Internet, RedHat reminds us that outdated security practices can undo even the most efficient DevOps initiatives. [4]

The top ten list of software vulnerabilities from the [Open Web Application Security Project \(OWASP\)](#) foundation has become one of the most relied-on tools for fostering collaboration between development, operations, and security teams. [5]

The US Air Force pioneered the DevSecOps Platform (DSOP) initiative, demonstrating that combining advanced DevOps and security practices can provide some of the most highly regulated organizations in the world with "plug and play" software factories and radically streamlined delivery processes.

Thanks to these contributions, security has become deeply ingrained in DevOps culture. As a result, *DevOps* and *DevSecOps* have come to mean the same concept for all practical purposes. Each implies a set of blended practices from multiple domains—development, operations, security, infrastructure, architecture, and so on throughout the value stream—that work together to enable collaboration, speed, quality, and safety.

Figure 2. Security is built into DevOps in SAFe

SAFe carries this sentiment forward, treating security as a primary concern. In SAFe, to say "DevOps" means "DevSecOps." Protecting customers, employees, citizens, soldiers, families, and businesses is not something we choose to do or not do in DevOps. It is simply in our DNA. As such, modern security practices shine through in many areas of SAFe, including the Big Picture, Framework guidance, courseware, assessments, [Extended SAFe Guidance](#) articles, and more.

These are just a few examples of how the DevSecOps movement has lifted DevOps to new standards of excellence.

Agile Release Trains (ARTs) are the primary value delivery construct in SAFe. Each ART has all the skills necessary to build and release the solution, including those

responsible for **Security**, Compliance, Quality Assurance (QA), Testing, and Verification and Validation (V&V). Each increment the ARTs builds assesses the viability of the current solution and its progress toward security, quality, and compliance, providing early feedback on the system's ultimate fitness for use. Second, specifications are created early and evolve in small batches, with faster feedback on decisions and the opportunity for continuous review and assessment. ARTs cannot implement security through *inspection; it must be built into the solution* during each iteration. Security testing should shift left to prevent vulnerabilities and be automated to increase the speed and accuracy of compliance.

Details

DevOps makes continuous delivery possible. Indeed, enterprises wishing to deliver value to customers and stakeholders continually should master the DevOps **mindset** and technical practices. These skills are critical in this era of constant digital disruption and innovation. Achieving continuous delivery, however, at scale, is not easy. SAFe's approach to DevOps helps enterprises navigate these complexities.

A Paradigm Shift

IT organizations worldwide are troubled by a core, chronic conflict: technology delivery processes rely on teams with *seemingly opposing* goals and incentives. [1] **Agile Teams** deliver changes quickly to keep pace with business needs. Operations regulate the flow of changes to maintain the stability of solutions that run the business. Security teams institute policies to prevent changes from introducing vulnerabilities that can cause data breaches.

Keeping pace requires a new delivery system — a "software factory" — which aligns teams and increases delivery speed while simultaneously increasing solution quality, security, and stability. Only then can the needs of customers and teams be predictably and effectively met.

Note: "Software factory" is an increasingly popular term for this new delivery system. In his SAFe [Community Contributions](#) article, Peter Vollmer, Distinguished Technologist at Micro Focus, describes a software factory as "standardized tooling and engineering services" that supports and enhances [the value stream]. [6]

These software factories are integrated sets of tooling, services, data, and processes that help move products through the plan, build, test, and release cycles. [7] The US Department of Defense (DoD) maintains a growing ecosystem of software factories, leveraging a common DevSecOps Platform (DSOP) to rapidly deliver specialized digital products and services. [8] Regardless of the term used to describe the system, enterprises leverage DevOps to achieve this level of sophistication in their value streams.

Unfortunately, most IT organizations do not natively support this kind of system. Their processes and policies are optimized to prevent frequent changes to production systems, not enable them. Therefore, a paradigm shift is needed. Just as Agile represents a paradigm shift in how *we work*, DevOps represents a similar shift in how *we build*. Leveraging DevOps to usher in a new way of building digitally-enabled solutions is the key to transforming outdated development life cycles into CDPs.

Continuous Learning and Experimentation

CDPs are the result of applying DevOps effectively to value streams. And value streams need to behave differently than they did in the old model because today's technology delivery objectives differ.

Enterprises must release features faster than ever to remain relevant in their markets. But out-deploying the competition is not the goal. Out-learning them is. And that learning comes from understanding the new functionality's value in the market. Since features have no value until released, enterprises must constantly build, measure, and learn to evolve digital solutions that quickly attract and retain customers. Figure 3 shows that SAFe's CDP operates as a closed-loop system that fosters rapid, low-risk experimentation and continuous learning about customers' needs, habits, and preferences.

Figure 3. The CDP as a Continuous Learning Loop

Figure 3. The CDP as a Continuous Learning Loop

This relentless learning and experimentation engine is starkly different than traditional delivery processes. Enabling it requires a different mindset, skills, and tools across the entire value stream. Large batches, siloed teams, handoffs, monolithic architectures, change review boards, politics, and heroics have no place here. Instead, this new system needs to be guided by shared values, cross-functional collaboration, objective measurements, automation, and modern technical practices.

Enter DevOps.

Figure 4 illustrates how DevOps enables the CDP. It does this by supplying the *mindset*, *practices*, and *tooling* required to foster rapid delivery and learning at every step.

Figure 4. DevOps enables the CDP

Figure 4. DevOps enables the CDP

At its core, DevOps is a mindset that guides behavior and decision-making throughout the value stream. SAFe's CALMR approach to DevOps embodies this mindset, is central to the figure above, and permeates all aspects of the CDP. DevOps technical skills, practices, and tooling evolve and sustain solutions directly. In SAFe, *practice domains* represent this knowledge within the inner rings of the CDP model shown in Figure 4.

Measuring and Managing DevOps Maturity

Measuring DevOps performance and tracking incremental progress are essential to building a thriving DevOps culture.

The SAFe DevOps Health Radar (Figure 5) is a tool that helps ARTs and Solution Trains optimize their value stream performance. It provides a holistic DevOps health check by assessing the maturity of the CDP's four aspects and 16 activities.

The Health Radar measures baseline maturity at any point in a DevOps transformation and help guide fast incremental progress.

DevOps Health Radar

Figure 5. The SAFe DevOps Health Radar

Figure 5. The SAFe DevOps Health Radar

Download the free DevOps Health Radar assessment [here](#). Agility Health, a Scaled Agile partner, also offers an [online version of this assessment](#).

More in the DevOps Series

Article 1: DevOps Home Page (this page)

Article 2: [A CALMR Approach to DevOps](#)

Article 3: [SAFe's DevOps Practice Domains](#)

Learn More

[1] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

[2] Accelerate © State of DevOps 2019. <https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>

[3] 2019 State of DevOps Report. <https://puppet.com/resources/report/2019-state-of-devops-report>

[4] What is DevSecOps? <https://www.redhat.com/en/topics/devops/what-is-devsecops>

[5] OWASP Top 10 Application Security Risks. <https://owasp.org/www-project-top-ten/>

[6] Accelerating Flow with DevSecOps and the Software Factory.
<https://www.scaledagileframework.com/accelerating-flow-with-devsecops-and-the-software-factory/>

[7] Why a Software Factory Is Key to Your DevOps Success.
<https://techbeacon.com/devops/why-software-factory-key-your-enterprise-devops-success>

[8] Software Factories. <https://software.af.mil/software-factories/>

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*The emphasis
should be on why
we do a job.*

ÑW. Edwards Deming

ART and Solution Train Backlogs

The ART Backlog is a Kanban system that is used to capture and manage the features and enablers intended to enhance the solution and extend its architectural runway.

Adjust Cookies Settings The Solution Train Backlog is a Kanban system that is used to capture and manage the capabilities and enablers intended to enhance the large solution and extend its

architectural runway.

The ART and Solution backlogs capture the Solution's upcoming [Features](#), Capabilities, and [Nonfunctional Requirements \(NFRs\)](#). Managing the backlog is a critical economic driver for trains and the portfolio. [Product Management](#) has responsibility for the ART Backlog, while [Solution Management](#) is responsible for the Solution Train Backlog. These backlogs are visualized and managed in Kanban systems, where features, capabilities, and enablers are captured, defined, evolved, and prioritized to ensure a continuous flow of value to [Customers](#).

Details

Product and Solution Management develop, maintain, and prioritize the ART and Solution Train backlogs. They actively collaborate with stakeholders, including Customers, [Business Owners](#), [Product Owners](#) (POs), [System](#) and [Solution Architects](#), and others (for example, RTEs/STEs) to discover the features and capabilities needed to advance the solution.

Figure 1 illustrates a view of the ART and Solution Train backlogs with their primary input sources:

- 1) [Portfolio Epics](#) split into features or capabilities
- 2) Capabilities and enablers arising from the Solution Train's local context
- 3) Solution Train capabilities split into features and enablers
- 4) Features and enablers created from the ART's local context

Figure 1. Input sources for the ART and Solution Train Backlogs

Figure 1. Input sources for the ART and Solution Train Backlogs

An effective solution must meet all its functional and Nonfunctional Requirements (NFRs). NFRs shown at the bottom of the backlog in Figure 1 serve as constraints or restrictions across the backlog, affecting the solution's design and performance. They are typically captured in ***acceptance criteria*** or as part of the definition of done (DoD). NFRs are persistent qualities, and ARTs often revisit them as part of the DoD.

Building and Refining the Backlog

Product and Solution Management take a continuous, flow-based approach to refine their backlogs, ensuring features and capabilities are ready for implementation within the appropriate level of discovery and risk. Refining the ART and Solution Train backlogs to ensure readiness often involves the following activities:

- Discovering new features and capabilities
- Reviewing and updating backlog item definitions, including developing acceptance criteria and benefit hypothesis
- Identifying the enablers required to support new features and capabilities
- Applying [Behavior-Driven Development](#) (BDD) techniques to help clarify features and capabilities or holding specification workshops
- Prioritizing the backlogs using [Weighted Shortest Job First \(WSJF\)](#) in collaboration with Business Owners, System Architects, POs, and other stakeholders, such as RTEs/STEs
- Briefing Agile Teams and stakeholders about upcoming features and capabilities for [PI Planning](#)
- Deleting aging and no longer relevant items

Refinement activities often occur during the PO Sync, where Product Management and Product Owners identify new backlog items, revise others, and remove obsolete items. A well-maintained backlog is a prerequisite for a successful PI planning event and execution.

Managing the Backlog with Kanban

The ART and Solution Train Kanban system facilitates the flow of [Features](#) and Capabilities through the [Continuous Delivery Pipeline](#). Figure 2 illustrates a *typical* ART Kanban with *example* policies and WIP limits governing each state. While this is a good starting point, the system should be adapted to fit the train's needs, including the definition of WIP limits and the specific policies for each state.

Figure 2. A typical ART or Solution Train Kanban system

Figure 2. A typical ART or Solution Train Kanban system

The following process states describe the example ART and Solution Train Kanban flow.

Funnel -All new big ideas are welcome here, typically expressed as Features or Capabilities. These ideas may indicate needed new functionality, enhancement of the existing system functions, or [Enablers](#).

Analyzing \triangleright New ideas that align with the Solution [Vision](#) and [Strategic Themes](#) are further explored by [Agile Teams](#) when they have available capacity. Analyzing includes Continuous Exploration activities (e.g., Customer Centricity, Design thinking) and collaboration to create one or more well-formed features (see backlog refinement above). The WIP limit for this state considers the availability of Product and Solution Management, the capacity of teams, and other subject matter experts.

Ready \triangleright The highest-priority features analyzed and approved by Product Management or Solution Management advance to this state. They are prioritized

with [WSJF](#) relative to the rest of the backlog and await implementation.

Implementing \triangleright Features are pulled into the Implementing state as teams start working on them. Teams implement them throughout the PI.

Validating On Staging \triangleright The *validating on staging* step has two sub-states, *In Progress* and *Ready*:

In Progress \triangleright Features that are implemented and ready for feedback get pulled into this state. Teams integrate and test them with the rest of the system in a staging environment and present them for approval.

Ready \triangleright Approved features move to this state's "Ready" buffer, where they are prioritized again using WSJF to await deployment.

Deploying To Production \triangleright The *deploying to production* step also has two sub-states, *In Progress* and *Ready*:

In Progress \triangleright Features get moved to production immediately in an automated Continuous Delivery environment or to the *in-progress* state for manual deployment when capacity exists.

Ready \triangleright ARTs that separate deployment from release move the item to the *ready* buffer of *deploying to production* to await release. In other cases, features are automatically released, and users can immediately access them. This state is WIP limited to avoid the buildup of deployed but not yet released ñ features.

Releasing \triangleright When there's sufficient value, market needs, and opportunity, features are released to some or all customers to evaluate their benefit hypothesis.

Done \triangleright After a feature has been released and evaluated, it moves to the *done* state. However, new work items may be created based on customer feedback.

Managing ART and Solution Train Epics

Some ART and Solution Train initiatives are too big to be completed in a single PI. These ART or Solution Train [Epics](#) are typically identified and managed in a separate Kanban system, as shown in Figure 3. Also, some portfolio epics may require splitting into ART and Solution Train epics to facilitate incremental implementation. While mainly a local concern, these epics may impact financial, human, and other resources that might be large enough to warrant a Lean business case, discussion, and financial approval from [Lean Portfolio Management \(LPM\)](#). Epics that exceed the portfolio epic threshold require review and approval. This is a critical [Guardrail](#) on budgetary spending.

The primary purpose of this Kanban system is to analyze and approve ART and Solution Train epics, splitting them into features or capabilities that will be further explored and implemented using the ART or Solution Train Kanban. These Kanban systems may not be required depending on how frequently ART and Solution Train epics occur in the local context.

Figure 3. A typical ART or Solution Train Epic Kanban

Figure 3. A typical ART or Solution Train Epic Kanban

The process states in the ART or Solution Train Epic Kanban are like the [Portfolio Kanban](#), for example:

Funnel Ⓜ All big initiatives are welcome in the “Funnel” state. There is no WIP limit.

Reviewing Ⓜ Subject Matter experts (SMEs) and stakeholders review the epics and prioritize them using [WSJF](#) and other criteria to determine which ones should move on for more in-depth exploration. If the item exceeds the portfolio epic threshold established by LPM, they move to the Portfolio Kanban. The WIP limit for this state considers the availability of SMEs.

Analyzing During this exploration state, SMEs and stakeholders do the following types of analysis:

- Refine size estimates, and WSJF relative to other epics
- Consider solution alternatives
- Identify possible Minimum Marketable Features (MMF) or Minimum Viable Products (MVPs)
- Forecast the costs, and identify technology, architectural enablement, and infrastructure using a Lean business case (described in the epics article)

Guided by analysis and insights, Product and Solution Management and Business Owners (and LPM where required) approve or reject the epics. Approved epics are split into features or capabilities and moved to the ART or Solution Train Kanban funnel, where they will be prioritized based on WSJF. WIP limits apply to the analyzing state.

Like the portfolio Kanban, ART and Solution Train epics typically require [Epic Owners](#) to help define, explore, and implement them.

Balancing Value Delivery and Solution Health with Capacity Allocation

Every train needs to balance the backlog of business features or capabilities with continuous investment in enablers to build and maintain the architectural runway, avoiding velocity reduction and technology obsolescence. Further, these enablers support exploring requirements and applying design thinking for future PIs, creating prototypes and models, and enhancing visibility into opportunities and problem areas.

The collaboration during WSJF prioritization is often sufficient to communicate concerns and arrive at a good balance of work. When that approach is insufficient, Product and Solution Management may work with Architects to apply capacity allocation to decide how much of the total effort the ART will reserve for each activity type for the upcoming PI. Figure 4 illustrates an example of capacity allocation.

Figure 4. Example capacity allocation for a PI

Figure 4. Example capacity allocation for a PI

While the agreed-to capacity allocation can persist for several PIs, it should be periodically reviewed and adjusted during backlog refinement in preparation for PI planning.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[3] Reinertsen, Don. *Principles of Product Development Flow: Second Generation*

Lean Product Development. Celeritas Publishing, 2009.

Last update: 14 March 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

- Weekdays: 9am to 5pm
- Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



If you only quantify one thing, quantify the Cost of Delay.

—Don Reinertsen

Weighted Shortest Job First

Weighted Shortest Job First (WSJF) is a prioritization model used to sequence work for maximum economic benefit. In SAFe, WSJF is estimated as the relative cost of delay divided by the relative job duration.

[Adjust Cookie Settings](#)

In a flow-based system, priorities must be continuously updated to provide the

best economic outcomes. In other words, job *sequencing produces the best results* rather than prioritization based on a theoretical return on investment for each job.

To that end, SAFe applies WSJF to prioritize backlogs. Backlogs are continuously prioritized based on a WSJF algorithm that uses relative user and business value, time criticality, risk reduction and/or opportunity enablement, and job size. WSJF also conveniently and automatically ignores sunk costs, a fundamental principle of Lean economics.

Details

In *Principles of Product Development Flow* [2], Reinertsen describes a model (WSJF) for prioritizing jobs based on the Cost of Delay. Simply put, CoD is the money lost by delaying or not doing a job for a specific time. It's a measure of the economic value of a job over time. For example, if implementing a prospective feature would be worth 100,000 per month, and there was a delay of three months, the total CoD would be 300,000. In the SAFe context, jobs are the [Features](#), [Capabilities](#), and [Epics](#) contained in their respective backlogs.

Jobs that can deliver the most value in the shortest duration provide the best economic return. As applied in SAFe, the WSJF model supports the economic principles of Lean product development flow:

- Taking an economic view
- Ignoring sunk costs
- Making financial choices continuously
- Using decision rules to decentralize decision-making and control
- If you only quantify one thing, quantify the Cost of Delay

Figure 1 shows the impact of applying Reinertsen's WSJF for prioritizing jobs to be done. The blue-shaded areas illustrate the total CoD in each case. The jobs with the highest WSJF deliver the best economic outcomes. As the figure shows, picking the next best job to do can have a dramatic financial impact.

Figure 1. Applying WSJF for prioritization delivers the best overall economics

Figure 1. Applying WSJF for prioritization delivers the best overall economics

Estimating the Cost of Delay

As described above, the calculation of WSJF assumes one can determine the CoD (numerator) in absolute financial terms per unit of time, and the job time can be estimated with some degree of accuracy. In practice, however, both numbers can be extremely difficult to estimate. With regard to the numerator, CoD is an estimate at best; it's hard for anyone to know the actual value of a new job (a new feature) that has yet to be delivered to market. But Agile teaches us how to quickly estimate on a *relative* basis. Since there are many jobs to be done in the backlog, simply use relative numbers to compare jobs.

The first task is to get the right stakeholders together and collectively estimate the Cost of Delay relative to other jobs in the backlog. Figure 2 illustrates the three primary components of CoD for any particular job.

Figure 2. Cost of Delay components

Figure 2. Cost of Delay components

Figure 3 shows the formula for the CoD. First, compare backlog items relative to each other using the same modified Fibonacci numbers as in Ôestimating poker.Õ Then the (relative) CoD is calculated as follows:

Figure 3. Calculating the relative Cost of Delay

Figure 3. Calculating the relative Cost of Delay

Estimating the Job Duration

The next item in the equation, the denominator of WSJF, is the job duration. This duration can also be challenging to determine, especially early on, when the available capacity and time needed for each job are unknown. In other words, before doing the work, it's hard to know who will be working on it, how many people can be engaged, and how long it will take. However, since larger jobs take longer to complete than smaller ones, **job size** can be used as a good proxy for the **duration**. (As an analogy, if I'm the only one mowing my lawn, and the front yard is three times bigger than the back yard, it will take three times longer to cut.)

Using job size as a proxy for duration results in a straightforward calculation for comparing jobs via WSJF, as Figure 4 illustrates.

Figure 4. A formula for relative WSJF

Figure 4. A formula for relative WSJF

A Note on Using Job Size as a Proxy for Duration

It's important to note that job size is not a perfect proxy for job duration. Let's consider two scenarios:

Suppose specialty skills are readily available, enabling a large job with a high value to be delivered more quickly than expected. In that situation, it may provide more value in a shorter period. (If three people can mow my large front lawn while I do the small backyard, these jobs will have approximately the same duration but not the same value.)

A small job may have a scarcity of resources, or dependencies might mean that a smaller job could take longer than a bigger job.

If either of these is the case, simply use the relative estimated duration and adjust accordingly. But rarely do we need to worry about these two exceptions. In most situations, fast, WSJF relative estimating is adequate. Since this is a flow-based system, minor errors in the selection are not that critical, as that next important job will rise to the top of the backlog soon enough.

Calculating WSJF

The actual calculation and prioritization are more straightforward than the explanation that brings us to this point. Compare jobs (three features, in this example) for each CoD component and job size using a simple table or spreadsheet (Figure 5). As with estimating stories, the modified Fibonacci sequence reflects higher uncertainty when the numbers become larger. Specific instructions follow:

1. Start by estimating the CoD parameters in columns 1,2,3, *one column at a time*, setting the smallest item to 1. Then determine the other jobs estimate relative to that job. Note: it's critical to ensure each column has a 1 representing the smallest item, as that normalizes the parameters against each other.
2. Sum each component to calculate the CoD in column 5.
3. Estimate the job size in column 6; again, giving the smallest job a 1.
4. Calculate the WSJF by dividing the CoD by job size.

The highest WSJF is the next most important job to do.

Figure 5. A table for calculating WSJF

Figure 5. A table for calculating WSJF

This model encourages splitting large jobs into smaller ones. Otherwise, critical big jobs might never get done. But that's just Agile at work. Since the implementation is incremental, a different job will be selected whenever continuing work on a big job doesn't rank well against its peers. Another advantage of SAFe's WSJF model is that the specific monetary elements of CoD components are unnecessary, significantly reducing complexity and the time spent on prioritization. Instead, each

job is compared relative to the other items from the *same backlog*. Since updated backlog estimates include only the remaining job size, frequent reprioritization means that the system will automatically ignore sunk costs. It will always pick the next best job based on current economic factors.

Applicability of WSJF

WSJF is a general algorithm that is particularly useful in flow-based systems where frequent reprioritization is a driver of economic value. But it doesn't make decisions; it is simply a reasoning tool for use by stakeholders who must ultimately do so.

As presented in this article, WSJF is particularly useful for prioritizing features and capabilities in the ART and Solution Train backlogs. This is because:

Features are the primary economic driver for trains; some investment in prioritization is warranted.

There aren't many features in flight at any one time; An ART backlog typically contains perhaps 100 or so features waiting for attention. It is generally straightforward to maintain rolling WSJF estimates over time.

Since features are critical to ART performance, Input from Product Management, Architects, Business Owners, affected teams, and other stakeholders is critically important. It's worth the effort to insist on a collaborative effort to prioritize ART features.

But it's not as well suited for use in a few other places, including:

Team backlog prioritization ▷ Stories are small, and there are a lot of them in flight. Here, the priorities are driven by the priorities of the features in the ART backlog that spawned the stories and local concerns. It just isn't worth spending time on multi-parameter analysis and discussion. And since the stories are small and implemented in an iteration, the denominator (duration) is less of a priority determinant.

As the sole determinant in Portfolio Epic Prioritization ▷ The second area where WSJF is useful—but not totally adequate by itself—is in the later portfolio backlog prioritization steps. In the early kanban steps, WSJF is quite useful and is called out as such in the [Portfolio Backlog](#) article. But since these are substantial investments, a simple comparison of a $\hat{O}2$ to a $5\hat{O}$ might have implications running in the millions or tens of millions of Euros. In this case, more time should be invested in the estimates called for in the Lean business case, including value or potential monetary returns and better-informed speculation on likely duration. For more on using WSJF for Epics, see the [Epic](#) article.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[3] Reinertsen, Don. *Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

Last update: 8 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

English

Scaled Agile Framework



People at work are thirsting for context, yearning to know that what they do contributes to a larger whole.

~Daniel Pink

Vision

The Vision is a description of the future state of the solution under development. It reflects customer and stakeholder needs and the features and capabilities

proposed to meet those needs.

The vision is aspirational and achievable, providing the broader context—*an overview and purpose*—of the [Solution](#) being developed. It describes the markets, customer segments, and end-user needs. The vision sets the boundaries and context for new [Features](#), [Nonfunctional Requirements \(NFRs\)](#), and other work.

The vision can apply to any level of SAFe, explaining why it's on the Spanning Palette. While the vision's focus is typically on the solution, a portfolio vision is also clearly relevant, reflecting how [Development Value Streams](#) will cooperate to achieve the [Enterprise](#) objectives. [Agile Release Trains \(ARTs\)](#) and [Agile Teams](#) may also define a vision to communicate their part in developing the solution.

Details

Few question the benefit of Lean-Agile's focus on near-term deliverables and fast value delivery, which favors deferring decisions until the last responsible moment and limiting Work in Process (WIP). It also avoids Big Design Upfront (BDUF), future-proofing architectures, and overly detailed plans. There is no substitute for a bias for action. (‘Let's build it, and then we'll know.’)

However, every individual contributor makes many decisions in the context of large solutions. Therefore, continuously developing, maintaining, and communicating the vision is critical to creating a shared understanding of ART's goals and objectives, especially as those ideas evolve due to ever-shifting market needs and business drivers.

Portfolio Vision

The [Portfolio Vision](#) sets a longer-term context for near-term decisions in a practical and inspirational way. (‘This is something worth doing.’) Understanding the longer-term view helps Agile Teams make more informed choices about functionality development in the short and long run.

The portfolio canvas is a critical input in developing the portfolio vision. One of the primary uses of the canvas is to record the current state of the portfolio. This canvas represents the portfolio's *as-is* state, enabling the organization's alignment on its structure, purpose, and status. The next step is to envision the future state, which helps define the vision for the portfolio.

Lean-Agile Leaders are responsible for setting the company's strategic direction and establishing the mission for the teams implementing that strategy. [Switch](#) calls

this view a destination postcard, as Figure 1 illustrates [1].

Figure 1. The portfolio vision is an enterprise-level ‘postcard from the future.’

Figure 1. The portfolio vision is an enterprise-level ‘postcard from the future.’

A portfolio vision exhibits the following characteristics:

Aspirational, yet realistic and achievable ▷ It must be compelling and somewhat futuristic, yet practical enough to be feasible over some meaningful timeframe

Motivational to engage others on the journey ▷ The vision must align with the [Strategic Themes](#), mission, and purpose

[Business Owners](#) (or C-level executives) typically present this longer-term view and business context during the [PI Planning](#) event. These leaders can inspire and align the teams, increasing engagement and fostering creativity to achieve the best results.

Solution Vision

For **Large Solutions, Product** and **Solution Management** are responsible for translating the portfolio vision into a Solution Vision, describing the reason and direction behind the chosen solution. Doing so requires specific questions to be asked and answered:

- What will this new solution do?
- What problems will it solve?
- What Features and benefits will it provide?
- For whom is this solution?
- What Nonfunctional Requirements will it deliver?

Inputs to the Solution Vision

Product and Solution Management work directly with Business Owners and other stakeholders to synthesize all the inputs and integrate them into a holistic and cohesive vision, as Figure 2 illustrates. These inputs include the following resources:

Figure 2. Solution vision input sources

Figure 2. Solution vision input sources

Customers Ⓜ [Customers](#) provide fast feedback and have intimate knowledge of what is needed

Strategic Themes Ⓜ The [Strategic Themes](#) provide direction and serve as decision-making filters

Portfolio Vision Ⓜ The [Portfolio Vision](#) gives the current state and the desired future state of the portfolio

Solution Context Ⓜ The [Solution Context](#) indicates how the solution interacts with the customer's context

Solution Train Backlog Ⓜ The [Solution Train Backlog](#) contributes direction and guidance to the vision

Solution Intent Ⓜ The [Solution Intent](#) contains some of the vision and is the destination for new elements

Architects Ⓜ The [System](#) and [Solution Architects](#) support the continuous evolution of the [Architectural Runway](#) and support current and near-term features.

Agile Teams Ⓜ Finally, and not to forget the obvious, the foremost experts in the domain are typically Agile Teams.

Product Owners Ⓜ The [Product Owners](#) continuously communicate emerging requirements and opportunities.

Capturing Vision in Solution Intent

Given the SAFe practice of cadence-based PI planning, vision documentation (various examples can be found in [2], [3], and [4]) is accompanied and sometimes replaced by rolling-wave vision briefings. These briefings provide routine, periodic presentations of the short- and longer-term vision for the ART. During PI planning for large solutions, stakeholders, such as [Solution Management](#), describe the overall *solution vision*, while [Product Management](#) provides the specific ART context.

The relevant elements of the vision, including details of the system's current system behaviors, are captured in the solution intent.

Roadmap View

Having a sense of direction is critical to planning and engagement. But unless there is some realistic plan for how teams intend to fulfill the vision, people won't know what they must do. The [Roadmap](#) fills that purpose. Figure 3 provides an example.

Figure 2. The roadmap is part of the vision

Figure 3. The roadmap is part of the vision

PI Planning Vision—The Top 10 Features

The roadmap is indeed helpful. But for action and execution, the immediate steps must be clear. Product and Solution Management are responsible for directing these next steps. In SAFe, this translates to a series of incremental steps forward, one PI and one feature at a time (Figure 4).

Figure 4. Vision is achieved one PI at a time via the top 10 features for the following PI

Product Management constantly updates feature priorities using Weighted Shortest Job First ([WSJF](#)). Then, during PI planning, they present the top 10 to the team. The team won't be surprised by the new list, as they have seen the vision evolve and are aware of the upcoming new features. Further, the [ART Kanban](#) is used to explore features' scope, benefit hypotheses, and acceptance criteria so they are sufficiently well-formed and vetted when they reach this boundary. Architect has already reviewed them, and various [Enablers](#) have already been implemented.

However, everyone understands that the top 10 is an input, not an output, to the planning process. Some ARTs will have more or less than ten features that they bring into PI planning based on the ART's velocity. What can be achieved in the following PI is subject to capacity limitations, dependencies, the knowledge emerging during planning, and more. Only the teams can plan and commit to a course of action summarized in the team [PI Objectives](#).

But these features are ready for implementation. And feature by feature, the ART marches forward toward accomplishing the vision. Solution Management presents a similar top ten [Capabilities](#) list during Pre-PI planning to align the ARTs within a Solution Train.

Learn More

[1] Heath, Chip, and Dan Heath. *Switch: How to Change Things When Change Is Hard*. Broadway Books, 2010.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[3] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.

[4] Leffingwell, Dean, and Don Widrig. *Managing Software Requirements*. Addison-Wesley, 2001.

Last update: 28 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[Usage and Permissions](#)

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*This person is a
needle in a haystack.
An almost
impossible
combination of
structured thinker
and visionary leader.*

— Tony Fadell, Build
[1]

Product

[Adjust Cookie Settings](#)

Find a Course:

Management

Go

Product Management is the function responsible for defining desirable, viable, feasible, and sustainable solutions that meet customer needs and supporting development across the product life cycle.

The role scales with the complexity of the [Solution](#). For some solutions, the Product Management function may be carried out by a single Product Manager. For others, a team of Product Managers may be required.

Details

Successful solutions progress through the four distinct stages of the product life cycle: introduction, growth, maturity, and decline. Product Management guides each solution through these stages, ensuring that maximum business value is achieved over the life of the solution.

Regardless of whether a solution is consumed internally or sold externally, Product Management is essential to its success. This article describes Product Management's multi-faceted role in SAFe and how it enables the continuous, sustained delivery of value.

Key Collaborations

Converting ideas into a steady stream of valuable products and services is a complex endeavor that requires ongoing coordination and communication across the organization. Product Management is integral to this process but does not have all the domain knowledge or capacity to execute it alone. Figure 1 illustrates the collaborative partnerships Product Management maintains to enable quick and efficient product development flow.

Figure 1 - Key Product Management collaborations

Figure 1. Key Product Management collaborations

Each branch in the diagram represents a set of relationships that Product Management leverages to guide solutions effectively through the product life cycle. The nature of these relationships is described below.

Align on outcomes ▷ Product Management ensures that solutions deliver tangible business value. They collaborate directly with [Customers](#), [Business Owners](#), and, when appropriate, [Solution Management](#) to understand the market forces, desired economic outcomes, and broader solution vision influencing product strategy.

Steer the ART ▷ Product Management collaborates frequently with [System Architects](#) and the [Release Train Engineer](#) (RTE) to guide the ART toward successful delivery. Together they form an essential leadership triad that maintains constant synergy between product strategy and implementation.

Evolve the solution ▷ Ultimately, it is up to [Agile Teams](#) to execute the product strategy by developing, delivering, and continuously enhancing solutions. Product Management synchronizes frequently with the [Product Owners](#) of those teams, providing them with accurate business context and receiving feedback on the feasibility of implementation.

Responsibilities

Product Management's responsibilities in SAFe fall into five main areas, as shown in Figure 2.

Figure 2. Product Management areas of responsibility

Figure 2. Product Management areas of responsibility

Each area of responsibility is described below, followed by additional guidance for ensuring that the Product Management function delivers maximum value to customers and the enterprise.

Exploring Markets and Users

Product Management continuously explores the [Solution Context](#), gathering qualitative and quantitative insights about market dynamics and user preferences. These insights inform both business strategy and technical strategy and produce the hypotheses that fuel the [Continuous Delivery Pipeline](#).

Conduct primary and secondary research Ⓛ Primary data answers specific questions about product-market fit in specific usage contexts. Secondary data reveals macro-level trends across broad cross-sections of the market. Product Management leverages both to inform overall product strategy and specific elements of product design.

Apply market segmentation Ⓛ Not all users are the same. They have different challenges, desire different features, and value products differently at different times. Product Management divides the user population into segments based on common characteristics and defines solutions for the most appealing segments.

Identify market rhythms and events Ⓛ Market rhythms are predictable, often seasonal, patterns in supply and demand that inform periodic release schedules. Market events, such as regulatory changes, patent expirations, and releases from competitors, occur more sporadically and require Product Management to plan for dynamic, out-of-cycle releases.

Understand end-user needs Ⓛ Ensuring that solutions deliver maximum business benefit requires a deep understanding of the needs of end users. Product Management employs iterative elicitation techniques such as [Lean UX](#), human-centered design (HCD), and journey mapping to evolve product strategy in alignment with these ever-changing needs.

Connecting with the Customer

Product Management engages directly with customers throughout the product life cycle. This ensures that the customer's needs are built into product strategy from the start and remain reflected in released solutions as the customer's needs change over time.

Adopt a customer-centric mindset Ⓛ Effective product management is driven by a [customer-centric](#) mindset in which the customer is placed at the center of every decision. Supported by the tools and techniques of Design Thinking, this mindset focuses the entire organization on creating desirable, viable, feasible, and sustainable solutions.

Empathize with the customer Ⓛ Solutions must deliver valuable experiences. This requires organizations to design each solution from the customer's perspective, putting preconceived ideas aside. Product Management leads the way by utilizing personas, empathy interviews, empathy maps, and related tools to capture and communicate customer wants and needs.

Apply design thinking Ⓛ [Design Thinking](#) is a holistic, iterative approach to ensuring that solutions are desirable, viable, feasible, and sustainable over the entire product life cycle. Product Management leverages design thinking practices and tools to fully understand the problems to be solved and to design the solutions that best address them.

Involve the customer continuously Ⓛ Agile product development is an iterative process that requires ongoing planning, doing, checking, and adjusting. Product

Management fuels this continuous feedback loop by involving customers frequently in [Continuous Exploration](#), [PI Planning](#), and [System Demos](#).

Defining Product Strategy, Vision, and Roadmaps

Product Management plays an integral role in bridging portfolio strategy and execution. Customer needs must be translated into solution concepts that can be delivered by Agile Teams and achieve measurable business outcomes.

Align strategy to business objectives \triangleright Product Management is responsible for aligning the product strategy, vision, and roadmap to the portfolio's [Strategic Themes](#) and for maintaining alignment with the [Portfolio Vision](#), [Lean Budgets](#), [Guardrails](#), and, when necessary, the solution vision. When assuming the role of [Epic Owner](#), a Product Manager also defines and governs the epic's Lean Business Case.

Establish equitable value exchange models \triangleright Following [SAFe principle #1](#) \triangleright Take an [economic view](#), Product Management identifies the specific value desired by customers from a solution in addition to the value the enterprise requires in return. An equitable exchange of value between the enterprise and its customers ensures solutions provide mutual, sustainable benefits.

Create and communicate a compelling vision \triangleright Product Management continuously refines and communicates the product vision to the ART. During each PI Planning session, Product Management presents the vision, highlighting prioritized features and relevant [Milestones](#). When appropriate, Product Management works with Solution Management during [Pre-PI Planning](#) to communicate the product vision to Solution Train stakeholders and calibrate it with the solution vision.

Manage flexible roadmaps \triangleright Product strategy and vision get codified in roadmaps that guide implementation. As Product Management adjusts product strategy and vision in response to changing business objectives and customer needs over the product life cycle, the roadmaps they influence are also adjusted.

Managing and Prioritizing the ART Backlog

Product Management supports the flow of work through the [ART Backlog](#), ensuring that it always reflects the most current needs of customers.

Guide Feature creation \triangleright [Features](#) are characteristics of a solution that fulfill specific customer needs and can be delivered within a single PI. Product Management ensures that features in the ART backlog contain clear benefit hypotheses and acceptance criteria.

Prioritize Features with WSJF \triangleright Because judicious selection and sequencing of features is a key economic driver for the ART, Product Management ensures that the backlog is reprioritized with [Weighted Short Job First](#) (WSJF) before each PI Planning session.

Accept Features \triangleright Acting on the customer's behalf, Product Management evaluates the completeness of features implemented through the ART backlog. This

final check examines a feature's implementation against its acceptance criteria and determines whether it contains enough business value to be released.

Support Architectural Runway While Product Management is not expected to drive technological decisions, they are expected to support the ongoing development and maintenance of [Architectural Runway](#). In collaboration with System Architects, they negotiate capacity allocations that balance the concentration of business and enabler features in the ART backlog.

Participate in ART events Product Management plays significant roles in PI Planning, [Inspect and Adapt](#) activities, biweekly System Demos, Solution Demos, and PI System Demos, actively imparting knowledge, collecting feedback, and addressing product-related issues. They may also participate as Business Owners, approving PI Objectives, assessing business value, and managing risks.

Delivering Value

Product Management leverages the Continuous Delivery Pipeline to release value with optimal market timing. Depending on the context, this can entail releasing multiple times per day, weekly, monthly, or whenever customer needs are in balance with the goals of the enterprise.

Collaborate throughout the value stream Product Managers with internal customers collaborate with [Development Value Stream](#) stakeholders and participants to ensure solutions deliver value to the enterprise. Product Managers with external customers collaborate with people and teams in the [Operational Value Stream](#) to deliver and maintain in-market solutions. When needed, Product Management also collaborates across the Solution Train to manage inter-ART dependencies and ensure timely integration of their products into large solutions.

Ensure product completeness Product management ensures that solutions meet a wide range of customer needs. Whole solutions are designed from the customer's perspective and comprise multiple features that together deliver complete, engaging end-user experiences.

Enable operations Product Management provides support and enablement to key functions in the operational value stream to ensure the full value of every release is realized. Marketing, sales, customer success, compliance, and channel partners, for example, receive assistance in preparing customers, stakeholders, and operations teams for product launches.

Release value on demand The Continuous Delivery Pipeline ensures that solution deployment is decoupled from release activities. This empowers Product Management, with input from stakeholders and customers, to [release](#) when the timing is optimal for the business.

Meet business goals An economically viable solution creates more value than it costs. Costs are relatively straightforward to measure, but value is often intangible. Moreover, enterprises assess value differently. Product Management is instrumental in defining the value of solutions within the enterprise context and tracking it to

ensure business goals are achieved.

Internal vs. External Customers

Product Management requires a deep understanding of the customer. Customers determine the value of every solution and, thus, are an integral part of the Lean-Agile development process.

SAFe defines two types of customers:

Internal customers are part of the enterprise. They receive solutions from one or more development value streams and leverage them in one or more operational value streams. For example, a team of underwriting managers at a bank may be internal customers of a credit scoring solution created by the IT department (Figure 3).

Figure 3. Product Management supports internal and external customers

Figure 3. Product Management supports internal and external customers

External customers are outside the enterprise. They purchase, license, or use solutions for their own benefit. Figure 3 depicts external customers that use

solutions in the operational value stream to submit loan applications and repay loans. The relationship between the enterprise and external customers can take the form of business-to-business (B2B), business-to-consumer (B2C), or business-to-professional (B2P) interactions.

It is common for product managers throughout the organization to work together in developing a final solution that is composed of several integrated solutions (Figure 4).

Figure 4. Teams of Product Managers working together to create a solution

Figure 4. Teams of Product Managers working together to create a solution

In this example, a mobile banking solution is delivered to external customers by a Mobile Banking Product Manager. The mobile banking app is built on a secure, compliant e-banking platform provided by a Digital Platform Product Manager. The core banking platform is built on scalable cloud infrastructure delivered by a Cloud Product Manager. Each oversees a solution that delivers a valuable product or service that is either consumed or further enhanced by the immediate customer. This scenario is especially common in the development of [Large Solutions](#).

Learn More

[1] Fadell, Tony. *Build: An Unorthodox Guide to Making Things Worth Making.* HarperCollins, 2022.

Last update: 7 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*The whole is greater
than the sum of its
parts.*

—Aristotle

System Team

The System Team is a specialized Agile team that assists in building and supporting the Agile development environment, including developing and maintaining the Continuous Delivery Pipeline. They may also support the integration of assets, end-to-end solution testing, DevOps mindset and practices, deployment, and release on demand.

Details

The growing complexity of modern software systems can dramatically affect the speed, quality, and flow of delivering value to customers. The most defining characteristic of today's infrastructure is that it's diverse to a fault [1]. It's easy for developers to lose focus, splitting their time between managing the issues associated with highly complex software development toolchains and delivering customer value.

Moreover, everyone has a cognitive limit to how much information they can hold in their minds at any moment. The primary reason to have a System Team is to increase the ART's speed, quality, and productivity without jeopardizing the team's ability to deliver new solution value. Agile Teams have an ever-increasing workload. Besides creating new solution value, teams are expected to continuously test, deploy, release, and even operate [Solutions](#). To be more efficient and have economies of scale, the ART may need the help of a System Team so that they can focus most of their time on solution development.

The System Team accelerates the ART's velocity and reduces the cognitive load on development teams by building and maintaining the Agile development environment infrastructure, for example:

The creation and maintenance of a [Continuous Delivery Pipeline](#) toolchain. This toolchain encompasses continuous integration, automating builds (including verification testing), deployment, and release.

Creating platforms and environments for development, user acceptance testing, solution demonstration, deployment, and production.

Facilitating the technical elements of collaborating with third parties like hosting and data service providers.

Ensuring the ART has an intentional architecture for the infrastructure and that teams use standard tools and [DevOps practices](#).

Responsibilities

Figure 1 illustrates the System Team's five primary areas of responsibility. However, it's important to note that while these responsibilities are highlighted here, the System Team and [Agile Teams](#) **share** them. Otherwise, the System Team can become a bottleneck, and the Agile Teams will not be fully capable of full end-to-end value delivery. The following sections describe each of these areas.

Figure 1. The System Team's primary areas of responsibility

Figure 1. The System Team's primary areas of responsibility

Note: The need for a dedicated System Team may decrease as the ART automates the CDP toolchain and learns how to deliver value independently and together. Over time, the knowledge from the System Team gets dispersed to the ART, and teams begin to have collective ownership of their broader responsibilities. For larger solutions, however, it's more likely that specialty expertise remains with one or more System Teams, as described in the section on Solution Trains below.

Building Development Infrastructure

A robust technical infrastructure supports high ART velocity, so the System Team may:

Create and maintain the CDP toolchain \triangleright this includes automating continuous integration, builds and verification testing, deployment, and release tooling

Create platforms and environments \triangleright creating environments for development, solution demos, and user acceptance testing in the cloud or on-premises

Facilitate the technical aspects of collaboration \triangleright act as the liaison with third parties, such as data, service, suppliers, or hosting providers

Supporting Solution Integration

Complex solutions also require that the System Team supports solution integration, which may involve the following activities:

Version management \triangleright determine and help maintain decisions and policies for version control and management

Create and run integration scripts \triangleright create and run solution-level build and integration scripts or manually integrate where automation does not yet exist

Attend various sync events \triangleright most work for the System Team comes from other Agile Teams, so various sync events (for example, PO, coach, architecture, or team syncs) helps them keep pace with the needs of the train before this work becomes a critical emergency

Assisting with End-to-End Testing

The System Team may also perform some testing duties that support Agile Teams:

Create newly automated test scenarios \triangleright Many teams may not have the infrastructure or knowledge to test functionality end-to-end. Systems Teams provide the environments to help teams create automated tests to support new or existing functionality. They may even write some of the more complex test scenarios.

Provide datasets \triangleright create re-usable test data and maintain it under version control.

Organize and manage test cases \triangleright organizes test cases designed by individual teams into ordered test suites, prioritize time-consuming tests, and improve testing infrastructure.

Perform manual testing \triangleright assists teams in performing manual testing for complex scenarios (such as end-end and exploratory tests) and some cross-system **Features** where automation is not yet feasible.

Conduct systems integration testing \triangleright assist the development team in verifying the integration between different systems. They test various integrations that form the end product, including integrations between software components and between software and hardware.

Create smoke tests \triangleright help create build verification tests to enable developers to verify their builds quickly and independently.

Test solution performance \triangleright helps create and run tests for NFRs (for example, security, reliability, performance, and usability) and assists **System** and **Solution**

[Architects](#) identify system shortfalls and bottlenecks.

Supporting System and Solution Demos

The ART conducts a [System Demo](#) at the end of each iteration, testing and evaluating the complete solution in a production-like environment (often Staging) to receive feedback from stakeholders and test the solution in an environment where the infrastructure and data closely match production. Similarly, the [Solution Demo](#) presents the combined development efforts of multiple ARTs and Suppliers to deliver value to [Customers](#) and other stakeholders.

The System Team supports ARTs or Solution Trains to plan, prepare and conduct System and Solution demos:

Planning demos \triangleright work with the Agile Teams to define what they will demo at the end of each iteration and prepare the needed environments and infrastructure.

Integrating assets \triangleright support the integration of assets and artifacts from Agile Teams.

Performing end-to-end solution demos \triangleright assist [Product Managers](#), [Product Owners](#), and others in running demos, predominantly end-to-end scenarios.

Set up and test the demo environment \triangleright prepare the technical environments needed to demo the new solution functionality reliably. Also, each new solution increment may require the System Teams to add extensions to the demo environment, including new interfaces, third-party components, simulation tools, and other environmental assets.

The System Team often helps trains make the necessary investments (in DevOps, CDP, etc.) to allow the system demo to happen in a timely cadence.

Facilitating Releases

The System Team supports ARTs and Solution Trains to plan, prepare, deploy, and release solutions. Activities can include the following:

Release governance \triangleright assist ART stakeholders with planning, managing, and governing solution releases. They also help Agile Teams ensure they meet any regulatory, industry, or other relevant standards and compliance.

Prepare for releases \triangleright package, and deploy solutions as part of DevOps and CDP activities. This includes using various techniques such as feature toggles to switch features on and off.

Release solutions \triangleright help the train release the solution to end users all at once, incrementally, or to different customer segments.

Stabilize and operate \triangleright collaborates with operations to ensure the solution works well from a functional and [Nonfunctional Requirements](#) (NFR) perspective.

Measure \triangleright helps the train instrument and quantify if the newly-released functionality

provides the intended value and the health and efficiency of the CDP.

Learn ☰ helps collect feedback from various sources and prepare for the next loop through the CDP.

The System Team in Solution Trains

For large, multi-ART [Development Value Streams](#) that need the constructs of the [Solution Train](#), System Teams are essential for supporting larger-scale integration and release challenges. Depending on the scope and complexity of the development value stream, there are three main patterns for structuring the System Team:

1. One System Team per ART that coordinates the solution integration and validation
2. A System Team for the Solution Train and individual System Teams for each ART
3. One System Team for the entire Solution Train, which can fulfill their responsibilities for all of its ARTs

The decision regarding which organizing pattern to use depends on the context. Factors include the ART structure within the Development Value Stream, the solution's [architecture](#), branching and integration policies, system testability, and development infrastructure.

Balancing Solution Integration and Testing Efforts

The System Team can never be the entire solution to the CDP tooling, infrastructure, and integration challenges. They need collaboration with other Agile Teams, aligned to a shared vision, to implement DevOps and the CDP. Efficient solution development requires sharing knowledge, best practices, and common infrastructure and tooling to help reduce complexity.

Maximizing ART velocity involves creating the optimal balance between Agile Teams and System Teams, as Figure 2 illustrates. With maturity and automation, the optimum point for integration responsibility shifts to the left, where the teams can reach higher degrees of autonomy and responsibility.

Figure 2. The optimum balance of integration effort

Figure 2. The optimum balance of integration effort

Learn More

- [1] O'Grady, Stephen. *The Developer Experience Gap*. RedMonk, 2020.
<https://redmonk.com/sogrady/2020/10/06/developer-experience-gap/>

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Inspection does not improve the quality, nor guarantee quality. Inspection is too late. The quality, good or bad, is already in the product. Quality cannot be inspected into a product or service; it must be built into it.

ÑW. Edwards
Deming

[Adjust Cookie Settings](#)

Built-In Quality

Built-In Quality is a set of practices to help ensure that the outputs of Agile teams in business and technology domains meet appropriate quality standards throughout the process of creating customer value.

Find a Course:

Go

Details

To support [Business Agility](#), enterprises must continually respond to market changes. The quality of the work products that drive business value directly determines how quickly the teams can deliver their solutions. Although work products vary by domain, they are likely to involve software, hardware designs, scripts, configurations, images, marketing materials, contracts, and other elements. Products built on stable foundations that follow standards are easier to change and adapt. Built-in quality is even more critical for large solutions, as the cumulative effect of even minor defects and wrong assumptions may create unacceptable consequences.

Building quality in requires ongoing training and commitment. But the benefits warrant the investment and include:

- Higher customer satisfaction
- Improved velocity and delivery predictability
- Better system performance
- Improved ability to innovate, scale, and meet compliance requirements

Built-in quality is linked to the fast flow of value described in [SAFe principle 6: Make value flow without interruptions](#). Accelerating problem discovery and taking corrective action occurs by shifting learning left on the timeline. Improved collaboration, workflow automation, more frequent delivery, and faster customer feedback support a quicker learning process.

SAFe applies Built-in Quality across five key domains. Each domain has a set of quality practices that vary from universally applicable generic practices to those specific to one or a few domains.

Figure 1 provides a consolidated view of Built-in Quality in SAFe.

Figure 1. Key domains and practices of Built-In Quality in SAFe

Figure 1. Key domains and practices of Built-In Quality in SAFe

The rest of this article describes the components of Figure 1 in deeper detail.

Built-in Quality Domains

Built-in Quality practices vary based on the domains in which they are applied. Despite the same intent behind the Built-in Quality approach to creating customer value, the actual practices reflect the intricacies of their environment and context. The following are the Built-in Quality domains in SAFe:

Business Functions

Business functions include marketing, sales, HR, finance, supply chain management, and other non-IT disciplines. Along with routine operations, each function also includes complex efforts requiring specific quality outputs for success. For example, creating a new marketing campaign or establishing new HR policies involve certain quality expectations.

Software Applications

Software is an essential contributor to business agility, the ability to scale the business, and better compete in the digital age. But seizing such opportunities requires maintaining predictable quality when delivering solutions.

IT Systems

IT infrastructure powers vast ecosystems of today's enterprise solutions landscape. The more complex the solutions, the more sophisticated the IT systems must be to sustain them. To support the reliable operation of the enterprise, IT systems require substantial quality standards and, therefore, proper quality practices.

Hardware

When used in computer technology, hardware typically refers to cables, monitors, integrated circuits, and other tangible elements of a computer system. But more generally, hardware refers to devices with concrete physical properties: mass, size, and matter. Examples include motors, gears, tools, chassis, cases, and simple or complex mechanisms. Due to their significantly higher cost of change, hardware systems require a unique approach to quality.

Cyber-physical Systems

Cyber-physical systems are complex systems wherein multiple physical elements are controlled by software algorithms. Examples include robots, aircraft, and automobiles. These are some of the world's most complex systems and often include intricate electrical, mechanical, optical, fluidic, sensory, and other subsystems. Their complexity and the high impact of failure emphasize the critical importance of quality in such systems.

Basic Agile Quality Practices

Basic Agile quality practices can be applied to work products in any domain. They have proven their worth and provide a common starting point for knowledge workers to understand and improve the quality attributes of the artifacts, work products, systems, and services that benefit themselves and their customers. A set of five SAFe Basic Agile Quality Practices are described in the sections below.

Shift Learning Left

Every development effort involves numerous unknowns that surface as development progresses and teams learn new facts. If the learning happens late in the process, underlying issues will significantly impact the solution, and significant rework and delays will result. However, if learning takes place much earlier—*or is shifted left*—problems reveal themselves sooner, enabling corrective action with minimum impact (Figure 2).

Figure 2. Late problem discovery vs. shift left

Figure 2. Shifting quality left reveals problems sooner

Shifting learning left does not simply mean that some actions take place earlier on the timeline but also that the structure of some of the basic processes is changed. For example, a test-first approach requires shifting away from conventional testing. Instead, tests are created whenever possible **before** the desired solution functions are implemented.

Pairing and Peer Review

Pair work describes a practice wherein two knowledge workers collaborate over the same asset in real time. Often, one serves as the driver, directly advancing the work product, while the other acts as the navigator, providing real-time evaluation and feedback. Team members switch roles frequently. Because the work product will contain each member's shared knowledge, perspectives, and best practices, pairing creates and maintains higher quality. As teammates learn from each other, the skillsets of the entire team rise and broaden. Additionally, peer review helps spot quality issues as one team member examines the work products of the other. Many governance processes around software, for example, mandate peer review as a compliance activity.

Collective Ownership and T-shaped skills

Collective Ownership is a quality practice where individual team members have the requisite skills and authority to update any relevant asset. This approach reduces dependencies between teams and ensures that any individual team member or team will not block the fast flow of value delivery. Any individual can add functionality, fix errors, improve designs, or refactor *because the work product is not owned by one team or individual*. Collective ownership is supported by quality standards that encourage consistency, enabling everyone to understand and maintain the quality of each component. Collective ownership is further enabled by ÔT-shaped skills. T-shaped skills characterize individuals who possess deep experience in one area but also have broad skills in other areas. T-shaped skills also represent the ability to work well with others.

Artifact Standards and definition of done

Assets created and maintained by the organization must adhere to standards that help ensure their value to the business. These standards may reflect how the artifacts are being built or what properties they must manifest. Standards are often unique to the specific organization and solution context, emerging gradually, validated frequently, and corrected by multiple feedback cycles. To productively maintain artifact standards, the teams must understand the motivations for their existence. Artifact design practices and the effective use of automation help facilitate standards. Enacting productive artifact standards involves applying a definition of done (DoD) Ô an essential way of ensuring that a work product is complete and correct. Each team, train, and enterprise should build a DoD that suits their needs.

Workflow Automation

Workflows tend to have many manual steps. Handoffs from one worker to another, searching for an asset of interest, and manual inspection of an asset to a standard are just a few examples. The fact is, all these manual steps are error-prone and cause delays in the process. Many of these tasks can be automated if the teams take the time to invest in a more automated pipeline that supports the activities. Automation provides substantial gains due to reduced execution costs and intrinsic adherence to standards. Of course, this can be done incrementally, and it often starts by putting a Kanban system in place and then noting steps that can be automated. Sometimes the first step is simply setting up automated notifications when an item changes state. Even simpler, many such systems are designed as true pull systems where the worker simply checks the system to see what work is available to them based on its state. In this case, the handoff is automatic and doesnÔt require separate communications overhead just to know the state of a work product.

Business Quality Standards

The above sections describe a set of five basic Agile quality practices that can be applied to every business domain. Virtually every aspect of business operations—accounting and finance, legal, sales, development, HR, marketing, operations, production, and more—is subject to internally or externally imposed quality standards, which are often linked to compliance requirements. Each business function produces specific outputs, which must satisfy quality standards relevant to that context.

No matter your business function, the steps to achieve quality with Agility include the following:

- Organize into Agile teams, get trained, and iterate.
- Define the standards and compliance policies for your function.
- Agree on the definition of done (DoD) for artifacts and activities for your workflow.
- Implement the basic Agile quality practices.
- Measure and learn. Specialize Agile quality practices further to your specific function.
- Improve relentlessly.

Agile Software Development Quality Practices

Software may well be the richest and best-defined area for applying Built-in Quality. This was driven by necessity, as software is exceedingly complex and intangible. You can't touch it or see it, so traditional approaches to inspecting, measuring, and testing are inadequate. If quality isn't built in endemically, then it's unlikely to exist at all. To address this new challenge, many new quality practices like those above were

inspired by Extreme Programming (XP), which has a zest for going fast

with quality. They have proven their worth and have now started influencing quality practices in other domains. The practices below apply well to software development, and we will describe them in that context, but they can be applied to other domains as well.

Continuous Integration

Building large-scale value requires knowledge workers to build the system in increments, resulting in frequent small changes. Each must be continually checked for conflicts and errors and integrated with the rest of the system to assure compatibility and forward progress. [Continuous Integration](#) (CI) provides developers with fast feedback (Figure 3). Each change is quickly built, integrated, and then tested at multiple levels. CI automates the process of testing and migrating changes through different environments, notifying developers when tests fail.

Figure 3. Continuous integration (CI) fosters system-wide quality

Figure 3. Continuous integration (CI) fosters system-wide quality

Continuous integration is vital within and across teams, allowing them to quickly identify and resolve issues in all parts of the codebase.

Test-first Practices

Agile teams operate in a fast, flow-based system to develop and release high-quality business capabilities quickly. Instead of performing most of the testing at the end, Agile teams define and execute many tests early and often as a part of their integration process. Tests are defined for small units of code using [Test-Driven Development](#) (TDD), for [Story](#), [Feature](#), and [Capability](#) acceptance criteria using [Behavior-Driven Development](#) (BDD), and for the feature or capability benefit hypothesis using [Lean UX](#) (Figure 4). Building quality in ensures that Agile development's frequent changes do not introduce new errors while enabling fast, reliable execution.

Figure 4. Test-first practices accelerate flow

Figure 4. Test-first practices accelerate flow

Refactoring

Constantly changing technology and evolving business objectives make it difficult to maintain and continually increase business value. However, two paths to the future exist:

Keep adding new functionality to an existing code base toward an eventually unmaintainable “throw-away” state

Continuously refactor the system to build a foundation for efficiently delivering the current business value as well as future business value

Refactoring, which improves the internal structure or operation of an area of code without changing its external behavior, is better. With continuous refactoring, the useful life of an enterprise’s investment in software assets can be extended substantially, allowing users to benefit from a flow of value for years to come. But refactoring takes time, and the return on investment is not immediate, so an allowance for time and effort must be part of capacity planning considerations. For

more, see the extended guidance article on [Refactoring](#).

Continuous Delivery

Continuous delivery provides the ability to release value to customers whenever they need it. This is accomplished by the [Continuous Delivery Pipeline](#) (CDP), which contains four aspects: *continuous exploration*, *continuous integration*, *continuous deployment*, and *release on demand*. The CDP enables organizations to map their current pipeline into a new structure and use relentless improvement to deliver value to customers. Feedback loops internally within and between the steps and externally between the customers and the enterprise fuel improvements. Internal feedback loops often center on *process* improvements; external loops often center on *solution* improvements. The improvements collectively create synergy, ensuring the enterprise is Ôbuilding the right thing, the right wayÕ and frequently delivering value to the market. Additionally, SAFe [DevOps](#) features crucial practice domains for establishing fast and reliable value delivery mechanisms.

Continuous delivery helps SAFe teams release on demand. Releasing with quality, however, requires a specific, scalable definition of done that helps ensure that the requisite quality is built in. Figure 5 shows an example:

Figure 5. An example of a scalable definition of done

Figure 5. An example of a scalable definition of done

To support security practices, teams generate a Software Bill of Materials (SBOM) for each release describing the commercial and open-source components and dependencies to ensure no vulnerabilities.

Agile Architecture

[Agile Architecture](#) is a set of values, practices, and collaborations that support a system's active, evolutionary design and architecture. It embraces the DevOps mindset, allowing a system's architecture to evolve continuously while simultaneously supporting the needs of current users.

Agile architecture supports Agile development practices through collaboration, emergent design, intentional architecture, and design simplicity. It also enables designing for testability, deployability, and changeability. Rapid prototyping, set-based design, domain modeling, and decentralized innovation, in turn, support Agile architecture.

The essential concept of [Architectural Runway](#) allows Agile teams and trains to provide effective enablement for future business capabilities and features while progressively validating underlying architectural assumptions.

IT Systems Quality Practices

Every modern enterprise depends on properly functioning IT systems for its business success. With more and more business workflows being powered by IT, ensuring the reliability, scalability, safety, and security of IT systems becomes increasingly important. It requires a robust approach to building quality into these systems. A sample of IT-specific quality practices is described below.

Infrastructure as Code

One of the critical challenges in ensuring the quality of IT ecosystems comes from defining and sustaining configurations consistently. Often representing hundreds or even thousands of environment parameters, configurations grow out of sync and cause problems in different parts of the enterprise's solution landscape. "Infrastructure as Code" is an approach to control those configurations programmatically and thus benefit fully from automation in defining, procuring, and maintaining configurations consistently and integrally. Containerization is an excellent enabler of Infrastructure as Code, as it permits applying programming interfaces to various aspects of the execution environment. Additionally, using "immutable infrastructure" – an approach where IT components are rebuilt whenever needed, rather than modified in production – forces the organization to explicitly control all changes to the environment by formally redefining them and redeploying the component that changed.

NFRs and SLAs

IT infrastructure must provide certain qualities to the execution environment to support the systems essential to business operation. These quality attributes include things such as security, reliability, performance, maintainability, and scalability (Nonfunctional Requirements or NFRs). Additionally, relevant Service-Level Agreements (SLAs), such as Mean Time Before Failure (MTBF) and Mean Time to Repair (MTTR), must be ensured. In SAFe, NFRs and SLAs are achieved

incrementally by early and continuous testing and timely corrective action. Ensuring that systems meet their NFRS and SLAs requires instrumentation and the proactive build and use of the architectural runway.

Telemetry and Monitoring

Responding to unanticipated loads, security attacks, hardware, software, and network failures, require a range of options, from downgrading or removing services to adding service capacity. Telemetry and logging capabilities allow organizations to understand and fine-tune their architecture and operating systems to meet intended loads and usage patterns. Effective monitoring requires that full-stack telemetry is active for all features deployed through the CDP. Monitoring ensures that issues with system performance can be anticipated or addressed rapidly in production.

Cybersecurity Standards

IT environments must meet increasingly stringent quality standards to protect against unauthorized access, use, disclosure, or destruction. The spectrum of activities to achieve comprehensive cybersecurity includes:

- Technology enablement (data encryption, streamlined identity management, etc.)
- Frequent testing and validation (audits, penetration testing, etc.)
- Training and proper habits for the workforce
- Testing of all new assets for various vulnerabilities
- Frequently review new vulnerability alerts against existing solution's SBOM for affected components and provide patches or hotfixes

Automated Governance

Recent advances in DevOps and related methods, practices, and tooling provide new opportunities for IT teams to automate governance. Automated governance replaces tedious, manual, and error-prone activities and specifically addresses security, compliance, and audit needs. For more on this topic, see the reference [2]: Investments Unlimited, A Novel about DevOps, Audit Compliance, and Thriving in the Digital Age.

Automation of configuration management, audit, security testing (during both build and deployment), and immutable infrastructure help reduce human error that can lead to system vulnerabilities.

Agile Hardware Engineering Quality Practices

Ensuring quality in hardware systems and components is complicated because the cost of change increases with time, and the impact of quality issues with hardware is high. This can include catastrophic field failure, recalls of volumes of manufactured products, and expensive field replacement or repair. This risk pressures organizations to effectively apply Built-in Quality practices while developing engineered hardware systems and subsystems. There are several techniques organizations use to ensure Built-in-Quality in hardware systems, which are described below.

Modeling and Simulation

In Agile, the goal is to build and learn as quickly as possible. Modeling and simulation in the virtual environment—and rapid modeling in the prototype environment—help shift learning left, as shown in Figure 6.

Figure 6. Shift learning left for hardware engineering

Figure 6. Shift learning left for hardware engineering

Analysis and simulation in digital models used in electrical and mechanical Computer-Aided Design (CAD) and MBSE (see below) can test changes quickly and economically. Digital twins combine multiple virtual models with data harvested from telemetry in the operational systems to improve the models and better predict how systems will behave in the future. The feedback loops in Figure 6 show how data from other environments validate and improve the digital environment. Some aerospace and automotive products even use model simulations for certification, substantially reducing the time and cost of changes.

Rapid Prototyping

The virtual environment cannot reveal all issues. Physical prototypes are a lower-cost substitute for real, Òbent metalÓ hardware. They provide higher-fidelity feedback, available only in a physical environment. Example prototype practices include:

Wood and other low-fidelity mockups

Breadboarding electrical components
3d-printed mechanical and electrical parts (PCBs, wiring harnesses)

Increasingly, additive manufacturing is used to lower the costs of rapid experimentation and prototyping. Additive manufacturing uses data computer-aided-design (CAD) software or 3D object scanners to direct hardware to deposit material, layer upon layer, in precise geometric shapes. As its name implies, additive manufacturing adds material to create an object. By contrast, when you create an object by traditional means, it is often necessary to remove material through milling, machining, carving, shaping, or other means.^[3]

Many organizations with the equipment and knowledge to print mechanical and electrical parts can produce and ship them in a single day. And parts made with additive manufacturing are now making their way into production.

Cyber-physical Systems Quality Practices

Cyber-physical systems require an organization to deal effectively with hardware components and the software that governs its behavior. Additionally, because such systems operate directly in the real world, the impact of quality issues can be significant and often subject to regulatory compliance.

Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE) is the practice of developing a set of related digital models that help define, design, and document a system under development. These models provide an efficient way to explore, update, and communicate system aspects to stakeholders while significantly reducing or eliminating dependence on traditional documents. By testing and validating system characteristics early with the model, they facilitate timely learning of properties and behaviors, enabling fast feedback on requirements and design decisions.

Frequent End-to-end Integration

In the software domain, continuous integration is the heartbeat of continuous

delivery: It's the forcing function that verifies changes and validates assumptions across the entire system. Agile teams invest in automation and infrastructure that builds, integrates, and tests every developer change, providing immediate feedback on errors.

Large, cyber-physical systems are far more challenging to integrate continuously because:

- Long lead-time items may not be available
- Integration spans organizational boundaries
- Automation is rarely end-to-end
- The laws of physics dictate certain limitations

Instead, *frequent* end-to-end integration addresses the economic tradeoffs of the transaction cost of integrating versus delayed knowledge and feedback (Figure 7).

Figure 7. Frequent end-to-end integration

Figure 7. Frequent end-to-end integration

The goal is frequent partial integration with at least one complete solution

integration for each PI.

Learn More

[1] https://en.wikipedia.org/wiki/Software_supply_chain

[2] Beal, Helen and Bill Bensing , Jason Cox , Michael Edenzon , John Willis.

Investments Unlimited: A Novel about Devops, Security, Audit Compliance, and Thriving in the Digital Age, IT Revolution Press, 2022

[3] <https://www.ge.com/additive/additive-manufacturing>

Last updated: 18 January 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



It is hard to imagine a more stupid or more dangerous way of making decisions than by putting those decisions in the hands of people who pay no price for being wrong.

—Thomas Sowell

Business Owners

Adjust Cookie Settings

Business Owners (BOs) are key ART stakeholders who have the primary business and technical responsibility for return on investment (ROI), governance, and compliance.

Business Owners are critical stakeholders who evaluate fitness for use and actively participate in ART events and solution development.

Details

Self-managing, self-organizing [Agile Teams](#) and ARTs are essential to the success of SAFe. This Lean-Agile way of working represents a significant change in the traditional management mindset. Leaders and management no longer need to supervise work directly or assign tasks. Instead, they lead and provide intent by establishing a mission and [Vision](#).

Business Owners may help teams with coaching and skills development but essentially decentralize execution authority to the ART. However, transformation to a Lean-Agile way of working does not relieve leaders and management of their ultimate responsibilities. They remain accountable for the organization's growth and its people, operational excellence, and business outcomes. SAFe defines the role of BOs, the key leaders who guide ARTs to optimal business outcomes.

Questions to identify Business Owners include:

- Who is ultimately responsible for business outcomes?
- Who can steer this ART to develop the right solutions?
- Who can speak to the technical competence of the solution now and in the future?
- Who should participate in planning, help eliminate impediments, and speak on behalf of development, the business, and the customer?
- Who can approve and defend a set of PI plans, knowing they will never satisfy everyone?
- Who can help ARTs coordinate efforts with other departments and organizations, spanning organizational boundaries?

The answers to these questions will help identify the BOs who will play a vital role in the ART's ability to deliver value. Also, consider the availability of these leaders and their personal traits. Will they be an excellent [Lean-Agile leader](#)? Are they interested in fulfilling this role?

It's best to start with the smallest possible Business Owner team and then add members if it becomes clear that someone with the necessary accountability, skill, knowledge, or expertise is missing. Ensure a good mix of both business-oriented and technical people. It's a reasonable expectation that membership in the BO

team will change as needs dictate.

Responsibilities

An effective Business Owner is active and involved, fulfilling their SAFe responsibilities daily, as illustrated in Figure 1.

Figure 1. Business Owner Responsibilities

Figure 1. Business Owner Responsibilities

While there is no precise guideline about who should be part of the Business Owner team, they often have the following roles or titles:

- General or line of business manager
- Product or Solution Managers
- Enterprise Architects
- C-level executives
- Operations executives

Senior engineering leaders
Customers (for bespoke solutions)

The following sections describe the Business Owner's duties, enabling them to fulfill their obligations while empowering Agile Teams and trains to do their best work.

Leading by Example

Business Owners are **Lean-Agile Leaders** who share accountability for the business value delivered by a specific ART. The most important and effective technique for driving the cultural change needed for the adoption of SAFe is for leaders to internalize and model the behaviors and mindsets of **Business Agility**. Such leaders inspire others to follow in their direction and to incorporate the leader's example into their development journey. To accomplish this, Business Owners:

Serve as an example of the new behaviors ☐ Live by the Lean-Agile principles and practices, modeling the new norms of expected behaviors for the ART and others to follow. They help address shortcomings in SAFe knowledge and experience.

Communicate the vision for SAFe adoption ☐ Frequently communicate the business need, urgency, and vision for change. BOs participate in developing the SAFe implementation plan, prioritizing the transformation backlog, and establishing the metrics for tracking the change progress for one or more ARTs.

Actively engage with the Lean-Agile Center of Excellence (LACE) ☐ Address problems that teams cannot resolve. Such issues are often beyond the span of control of the LACE. For example, they may require facility changes, funding, hiring, and purchasing authority.

Address the concerns of people who resist the change ☐ Exhibit empathy and compassion, address people's fears and worries, and resolve problems quickly and effectively to help overcome the resistance that may block the change.

Act as change agents ☐ Communicate passionately, sincerely believe, and illustrate their commitment to the future change vision. When people see leaders' behaviors modeling those required by change, they become change advocates, aligning with the new behaviors more quickly. BOs do not tolerate unacceptable behavior and inspire those who resist or fear the change with mission and vision. BOs help people understand the new way of working and how it will benefit them, other ART members, and the organization. These leaders assure people by committing to adapting roles, practices, and processes for the overall good of the organization and ART.

A lack of psychological safety at work can have significant business consequences. When people don't feel comfortable talking about things that aren't working, the organization is not equipped to prevent failure. After all, no one can fix a secret.

This fear often leads to disengaged employees and the *opportunity cost* to leverage the strengths of all its talent. People need to feel comfortable speaking up, asking naive questions, experimenting and failing with new ways of working, and disagreeing with changes to create and implement ideas that make a real difference.

In contrast, when employees are engaged, they adopt the organization's vision, values, and purpose. They become passionate contributors, innovating problem solvers, and dependable colleagues.

Engaging with LPM

While LPM is operated by executives responsible for business outcomes, Business Owners are often critically engaged in the process. Some Business Owners may serve as LPM executives, but most are involved to some extent in activities such as:

Strategy and Investment Funding □ Help ensure the portfolio and individual value streams are aligned and funded to create and maintain the solutions needed to meet business targets.

Agile Portfolio Operations □ Business owners are responsible for helping value streams and ARTs get the right thing out the door to their customers. They may also directly or indirectly support the LACE and foster **Communities of Practice (CoPs)** within their domain of concern.

Lean Governance □ Business Owners are directly engaged in ART backlog prioritization and value stream economics. They also help provide oversight and decision-making of spending, audit, compliance, forecasting expenses, and measurement for their value streams.

Occasionally serve as Epic Owners □ On occasion, Business Owners may serve as initial Epic owners for initiatives that benefit from their domain knowledge, experience, and authority.

Participatory Budgeting □ Business Owners actively assist LPM in allocating the total portfolio budget to its value streams.

Aligning Priorities and PI Planning

Business Owners are responsible for understanding and refining the **Strategic Themes** that influence ARTs. They have knowledge of the current **Enterprise, Portfolio, and Value Stream** context, and they're involved in driving or reviewing the solution vision and **Roadmap**. The continuous involvement of BOs during the PI serves as a critical **Guardrail** for the ART's budgetary spending. Aligning priorities and PI planning usually involve the following activities:

The time before PI planning is a busy period for Business Owners. Responsibilities include:

Provide input to backlog refinement ☰ Participate in activities to align the backlog with the portfolio's strategic themes

Ensure that business objectives are understood ☰ Ensure that the business objectives are agreed to by key stakeholders of the train, including the [Release Train Engineer \(RTE\)](#), [Product Management](#), [System Architects](#), and other BOs

Prepare to communicate the business context ☰ Prepare to describe the business's current state, the [Portfolio Vision](#), and their perspective on how effectively existing solutions address current customer needs.

The importance of the Business Owner's role during PI planning cannot be overstated. Activities include:

Present the business context and Vision ☰ Share the business context during the defined PI planning agenda timebox. This context may include the state of the business, market rhythms, milestones, and significant external dependencies, such as those of [Suppliers](#).

Actively engage during critical ART PI Planning activities ☰ Participate in draft plan reviews, assign business value to team PI objectives, and approve final plans.

Review draft and final plans ☰ Understand the bigger picture and determine if the team's objectives fulfill the current business objectives when taken together. They ask powerful questions and ensure alignment on solution intent.

Watch for significant external commitments and dependencies ☰ Foster the management of dependencies and support their reduction or elimination.

Actively circulate during planning ☰ Communicate business priorities to the teams and maintain agreement and alignment among the stakeholders regarding the key objectives of the train.

Participate in the management review and problem-solving ☰ Business Owners are critical stakeholders in this problem-solving meeting. They review and adjust the scope, resolve problems, and compromise as necessary.

Participate in Solution Train planning ☰ If applicable, BOs participate in [Pre-Planning](#), helping ARTs adjust their plans and providing support during the [Coordinate and Deliver](#) activities.

Moreover, when Business Owners assign planned business value during PI planning, it offers an essential face-to-face dialogue between teams and their most important stakeholders, the BOs. This activity is an opportunity to develop personal relationships between Agile Teams and BOs, identify common concerns that require mutual commitment, and better understand the business objectives and their value. Figure 2 provides an example of one team's PI objectives and the Business Value (BV) assigned by BOs.

Business Owners use a scale of one (lowest) to ten (highest) and will typically assign the highest values to the customer-facing objectives. However, they should also seek the advice of technical experts who know that architecture and other concerns will increase the team's velocity in producing future business value. So placing suitable business value on [Enablers](#) helps drive velocity and demonstrates their commitment to addressing the team's legitimate technical challenges.

Figure 2. An example of a team's PI objectives with assigned business value

SAFe customers often ask, why doesn't the BV use Fibonacci numbers? The answer is simple: the 10 to one scale is a range of numbers everyone understands, reducing friction and miscommunication between business-oriented BOs, technical members, and ART stakeholders. The simplest way to start is to assign a ten to the highest individual objectives, typically "fixed" commitments or must-have items, and then scale down from there. Giving many PI objectives a ten (for one team) indicates a lack of objective prioritization. It effectively abrogates prioritization to the team without the benefit of the BO's knowledge and experience.

Realizing Business Outcomes

The Business Owner's job is not complete when PI planning is done. They have an ongoing role in helping ensure the success of solution delivery. Business Owners typically:

- Maintain alignment** □ Actively maintain alignment between the business and development as priorities and scope inevitably change.

Help validate the definition of MVPs ☰ Guide pivot-or-persevere decisions for ART or Solution Epics based on the delivery of the MVP.

Attend the System and Solution Demos ☰ Actively engage in the system and solution demos to understand progress and provide feedback.

Attend Agile team events ☰ Attend team events such as [Iteration Planning](#), Review, and [Retrospectives](#) as needs dictate.

Actively address impediments ☰ Help resolve impediments that escalate beyond the authority of the trainðs leaders and stakeholders.

Participate in Release Management ☰ Serve as critical stakeholders in release governance (described in the [Release on Demand](#)) and determine when the solutions are released. Specifically, they focus on scope, quality, deployment options, release, and market considerations.

Sponsoring Relentless Improvement

The [Inspect and Adapt \(I&A\)](#) event is a cadence-based opportunity for the whole ART to reflect on progress and identify the systemic impediments theyðre facingÑmany of which require the BOðs involvement. During the event, BOs assess the actual value achieved versus the plan and participate in the I&Aðs problem-solving workshop. Moreover, Business Owners are Lean-Agile Leaders who:

Continually focus on eliminating waste and delays ☰ Foster the adoption of Principle #6, [Make value flow without interruption](#), including [Value Stream Management](#), the eight flow accelerators, and the six flow measurements (*flow distribution, velocity, time, load, efficiency, and predictability*).

Eliminate demotivating policies and procedures ☰ Actively participate in the I&Aðs problem-solving workshop to identify and eliminate systemic issues, policies, and processes that are not aligned with the Lean-Agile Mindset and are not within the scope of the ARTðs control.

Inspire and motivate others ☰ Effectively communicate (frequently) *why* change is needed and do so in ways that inspire, motivate, and engage people to buy into the change with a sense of urgency.

Create a generative culture that highly values relentless improvement ☰ Model the right behaviors to help transform the culture from pathological (negative, power-oriented) and bureaucratic (negative, rule-oriented) to a positive, performance-oriented culture, which is required for the Lean-Agile mindset to flourish.

Provide the time and space for teams to innovate ☰ Foster the use of IP iterations to provide a regular, cadence-based opportunity for teams to work on innovation, improvement activities, and learning that are often difficult to fit into a continuous, incremental value delivery pattern.

Help drive investment in the continuous delivery pipeline ☰ Supports process and infrastructure enhancements to the [Continuous Delivery Pipeline](#) to improve the

responsiveness of the ART and the quality of its solutions.

It cannot be emphasized enough: Active participation of Business Owners is critical to the SAFe enterprise.

Last update: 6 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

[Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



*100% utilization
drives
unpredictability.*

—Don Reinertsen

Innovation and Planning Iteration

The Innovation and Planning (IP) Iteration is a unique, dedicated iteration that occurs every PI. It provides an estimating buffer for meeting PI Objectives and dedicated time for innovation, continuing education, PI Planning, and Inspect and Adapt (I&A) events.

[Adjust Cookie Settings](#)

SAFe has an intense focus on continuous customer value delivery, and people are

busy working on the [Features](#) they committed to during [PI planning](#). Every Iteration counts and the teams are mostly “heads down,” delivering near-term value. One iteration after another, the [Solution](#) advances in the market. The attention to solution delivery is intense and unrelenting.

Of course, a focus on one thing—delivery—can lead to a lack of focus on another—innovation. Given the constant urgency for delivery, there’s a risk that the *tyranny of the urgent* [1] will override any opportunity to innovate. To address this, SAFe provides dedicated Innovation and Planning iterations.

Details

IP iterations provide a regular, cadence-based opportunity for every [PI](#) for teams to work on activities such as innovation and learning that are difficult to fit into a continuous, incremental value delivery pattern. Time is also needed for Inspect & Adapt, PI Planning, and the final preparations for cadence-based ART events. IP iterations fulfill another critical role by providing an estimating buffer for meeting [PI objectives](#) and enhancing the predictability of PI performance.

The benefits of consistently well-planned and executed IP iterations include the following:

- Better predictability and flow
- Increased employee engagement and reduced burn-out
- Greater agility and resilience
- Increased competitive advantage

The remainder of this article provides the specific activities that should be planned and supported during the IP iteration to maintain the health, proficiency, and effectiveness of the individuals and teams on the ART.

Allow Time for Innovation

Innovation culture is one of the [Continuous Learning Culture](#) dimensions, but finding time for innovation and change amid delivery deadlines can be difficult. To this end, many enterprises use IP iterations for research and design activities such as hackathons. There are two simple rules for hackathons:

- People can work on whatever they want, with whomever they want, so long as the work reflects the mission of the company
- The teams demo their work to others at the end of the hackathon

Hackathons allow the talented, creative individuals on the ART to explore their innovative ideas outside the usual constraints of their regular backlog and team construct. The learnings from hackathons routinely make their way into [ART Backlogs](#) and help drive innovations that directly benefit the business. They're fun, too!

Dedicate Time to PI Events

Performing the I&A and PI planning during the IP iteration avoids a reduction in the velocity of the regular iterations. More importantly, since these events are held on a regular cadence and can be scheduled well in advance, their occurrence is better guaranteed.

Also, some just-in-time, last-responsible-moment ART and Solution Train backlog refinement and feature and capability elaboration during this period can significantly increase the productivity of the upcoming planning session.

Integrate the Complete Solution

The PI [System Demo](#) occurs at the end of each PI. It is the integrated presentation of the work of all teams on the train, done in a staging environment, which emulates production as closely as possible. For ARTs that are part of a [Solution Train](#), the PI system demo feeds into the aggregate [Solution Demo](#), which also takes place during the IP Iteration. It's a more structured and formal affair, as it demonstrates the accumulation of all the features and capabilities developed throughout the entire PI for a Solution Train.

When a solution includes hardware (and other components), it's harder to integrate end-to-end continuously, and full integration may be feasible only during the IP iteration. In these cases, it's just common sense to plan for that.

Figure 1. Ensured integration points

Figure 1. Ensured integration points

However, the IP iteration should not be the only attempt to integrate the assets into the system. Full or partial integration happens throughout the PI (Figure 1), with a total solution integration occurring at least once per PI. This approach validates the assumptions early enough to respond to significant problems and risks within the PI.

Advance Development Infrastructure

Lean delivery puts increased pressure on the development infrastructure: new continuous integration environments require provisioning, new test automation frameworks must be implemented and maintained, Agile project management tooling must be adopted, upgrading or enhancing cross-team and train communications systems, and the list goes on. The improvement stories often come from the team's [Iteration Retrospective](#) or [Enablers](#).

We all understand that we must sharpen our tools from time to time; Agile teams

are no different. Indeed, they have an even higher dependency on their working environments, so time must be spent continuously improving them. It's often more efficient to improve infrastructure or perform a migration at a time when the teams aren't in the midst of critical work.

Enable Continuous Learning

Employees at every level are lifelong learners. Changes in technology, as well as changes to method and practice, are routine; opportunities for continuing education, however, are far less frequent. Also, the initial move to Lean-Agile requires many new techniques and skills, including:

- Feature and [Story](#) writing
- [Building in quality](#)
- Automated testing
- Collective ownership
- Understanding and applying flow
- [Agile Architecture](#)
- [Continuous Integration](#)
- Pairing and peer review
- Mastering [Product Owner](#) and [Scrum Master/Team Coach](#) roles
- Team building

It isn't easy for practitioners to keep their technical skills current. New technologies are being introduced more frequently than ever, which requires regular upskilling of the workforce. Investing in people who can work across multiple systems, domains, and languages creates a "T-shaped" (deep skill in one area, working knowledge in many other areas) and even an "E-shaped" (deep skill in more than one area) workforce. This expansion of the skills of the workforce gives the organization maximum agility and flexibility to deliver the most important backlog items. However, it isn't easy to find time for this growth alongside the drive to provide new features constantly. IP iterations are a perfect time for this investment.

Making time for continuing education gives teams and leaders a welcome opportunity to learn and master new knowledge and skills. ARTs can use the IP iteration to launch and support [Communities of Practice](#) devoted to a broad range of topics. The net results benefit both the individual and the enterprise: employee mastery, job satisfaction, and velocity increase, and time-to-market decreases.

Use the Built-In Estimation Buffer As Needed

Lean flow teaches us that 100 percent utilization drives unpredictable results [2]. Planning everyone to total capacity does not allow people the ability to flex when problems inevitably occur. The result is unpredictability and delays in value delivery. As a countermeasure, the IP iteration offers a guard band (or buffer) to prevent unfinished work from the current PI from carrying over to the following PI.

During PI planning, the ART does not plan features or stories for the IP iteration, providing a buffer (extra time) for the teams to adapt to unforeseen events. This buffer can offset delays resulting from dependencies, and other issues, increasing their ability to meet [Team and ART PI Objectives](#). It substantially increases the predictability of the outcomes, which is extremely important to the business. However, routinely using that time for completing work is a *failure pattern*. Doing so defeats the primary purpose of the IP iteration, and innovation will likely suffer. Teams must ensure that this estimating guard band does not merely become a crutch.

A Sample IP Iteration Calendar

IP iterations take on a somewhat standard schedule and format. Figure 2 provides an example IP iteration calendar for an ART.

Figure 2. Example calendar for an IP iteration

Figure 2. Example calendar for an IP iteration

Learn More

- [1] Hummell, Charles E. *Tyranny of the Urgent*. IPV Booklets, 2013.
- [2] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.
- [3] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 6 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Kaizen is about changing the way things are. If you assume that things are all right the way they are, you can't do kaizen. So change something!

—Taiichi Ohno

Inspect and Adapt

[Adjust Cookie Settings](#)

The Inspect and Adapt (I&A) is a

Inspect & Adapt: Overview

significant event held at the end of each PI, where the current state of the Solution is demonstrated and evaluated. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop.

The Agile Manifesto emphasizes the importance of continuous improvement through the following principle: ÔAt regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.Ó

In addition, SAFe includes Ôrelentless improvementÓ as one of the four SAFe [Core Values](#) as well as a dimension of the [Continuous Learning Culture](#) core competency. While opportunities to improve can and should occur continuously throughout the PI (e.g., [Iteration Retrospectives](#)), applying some structure, cadence, and synchronization helps ensure that there is also time set aside to identify improvements across multiple teams and [Agile Release Trains](#).

Details

All ART stakeholders participate along with the [Agile Teams](#) in the I&A event. The result is a set of improvement backlog items that go into the [ART Backlog](#) for the next PI Planning event. In this way, every ART improves every PI. A similar I&A event is held by [Solution Trains](#).

The I&A event consists of three parts:

1. PI System Demo
2. Quantitative and qualitative measurement
3. Retrospective and problem-solving workshop

Participants in the I&A should be, wherever possible, *all* the people involved in building the solution. For an ART, this includes:

The Agile teams
[Release Train Engineer \(RTE\)](#)
[System and Solution Architects](#)
[Product Management, Business Owners](#), and other stakeholders

Additionally, [Solution Train](#) stakeholders may also attend this event.

PI System Demo

The PI [System Demo](#) is the first part of the I&A, and it's a little different from the regular system demos after every iteration. This demo shows all the [Features](#) the ART has developed during the PI. Typically the audience is broader; for example, [Customers](#) or [Portfolio](#) representatives are more likely to attend this demo.

Therefore, the PI system demo tends to be a little more formal, and extra preparation and setup are usually required. But like any other system demo, it should be timeboxed to an hour or less, with the level of abstraction high enough to keep stakeholders actively engaged and providing feedback.

Before or as part of the PI system demo, [Business Owners](#) collaborate with each Agile Team to score the actual business value achieved for each of their [Team PI Objectives](#), as illustrated in Figure 1.

The *achievement score* is calculated by separately totaling the business value for the *plan* and *actual* columns. The uncommitted objectives are not included in the total plan. However, they are part of the total actual. Then divide the planned total by the actual total to calculate the achievement score illustrated in Figure 1.

Figure 1. Scoring actual business value for each team

Figure 1. Scoring actual business value for each team

Quantitative and Qualitative Measurement

In the second part of the I&A event, teams collectively review any quantitative and qualitative metrics they have agreed to collect, then discuss the data and trends. In preparation for this, the RTE and the [Solution Train Engineer](#) are often responsible for gathering the information, analyzing it to identify potential issues, and facilitating the presentation of the findings to the ART.

Each team's planned vs. actual business value is rolled up to create the ART predictability measure, as shown in Figure 2.

Figure 2. ART predictability measure is rolled up from each team's planned vs. actual business value

Reliable trains should operate in the 80-100 percent range; this allows the business and its external stakeholders to plan effectively. (Note: Uncommitted objectives are excluded from the planned commitment. However, they are included

in the actual business value achievement, as can also be seen in Figure 1.)

Retrospective

The teams then run a brief (30 minutes or less) retrospective to identify a few significant issues they would like to address during the problem-solving workshop. There is no one way to do this; several different Agile retrospective formats can be used [3].

Based on the retrospective and the nature of the problems identified, the facilitator helps the group decide which issues they want to tackle. Each team may work on a problem, or, more typically, new groups are formed from individuals across different teams who wish to work on the same issue. This self-selection helps provide cross-functional and differing views of the problem and brings together those impacted and those best motivated to address the issue.

Key ART stakeholders—*including Business Owners, customers, and management*—join the retrospective and problem-solving workshop teams. The Business Owners can often unblock the impediments outside the team’s control.

Problem-Solving Workshop

The ART holds a structured, root-cause problem-solving workshop to address systemic problems. Root cause analysis provides a set of problem-solving tools used to identify the actual causes of a problem rather than just fixing the symptoms. The RTE typically facilitates the session in a timebox of two hours or less.

Figure 3 illustrates the steps in the problem-solving workshop.

Figure 3. Problem-solving workshop format

Figure 3. Problem-solving workshop format

The following sections describe each step of the process.

Agree on the Problem(s) to Solve

American inventor Charles Kettering is credited with saying that “a problem well stated is a problem half solved.” At this point, the teams have self-selected the problem they want to address. But do they agree on the details of the problem, or is it more likely that they have differing perspectives? To this end, the teams should spend a few minutes clearly stating the problem, highlighting the “what,” “where,” “when,” and “impact” as concisely as possible. Figure 4 illustrates a well-written problem statement.

Figure 4. Example problem statement

Figure 4. Example problem statement

Perform Root Cause Analysis

Effective problem-solving tools include the fishbone diagram and the Ô5 Whys.Õ Also known as an [Ishikawa Diagram](#), a fishbone diagram is a visual tool to explore the causes of specific events or sources of variation in a process. Figure 5 illustrates the fishbone diagram with a summary of the previous problem statement written at the head of the Ôfish.Õ

Figure 5. Fishbone diagram with primary sources identified

Figure 5. Fishbone diagram with primary sources identified

For our problem-solving workshop, the main bones often start with the default categories of people, processes, tools, program, and environment. However, these categories should be adapted as appropriate.

Team members then brainstorm causes that they think contribute to solving the problem and group them into these categories. Once a potential cause is identified, its root cause is explored with the [5 Whys](#) technique. By asking ÔwhyÕ five times, the cause of the previous cause is uncovered and added to the diagram. The process stops once a suitable root cause has been identified, and the same process is then applied to the next cause.

Identify the Biggest Root Cause

Pareto Analysis, also known as the 80/20 rule, is used to narrow down the number of actions that produce the most significant overall effect. It uses the principle that 20 percent of the causes are responsible for 80 percent of the problem. ItÕs

beneficial when many possible courses of action compete for attention, which is almost always the case with complex, systemic issues.

Once all the possible causes-of-causes are identified, team members then cumulatively vote on the item they think is the most significant factor contributing to the original problem. They can do this by dot voting. For example, each person gets five votes to choose one or more causes they think are most problematic. The team then summarizes the votes in a Pareto chart, such as the example in Figure 6, which illustrates their collective consensus on the most significant root cause.

Figure 6. Pareto chart of probable causes

Figure 6. Pareto chart of probable causes

Restate the New Problem

The next step is to pick the cause with the most votes and restate it clearly as a problem. Restating it should take only a few minutes, as the teams clearly understand the root cause.

Brainstorm Solutions

At this point, the restated problem will start to imply some potential solutions. The team brainstorms as many possible corrective actions as possible within a fixed timebox (about 15Ð30 minutes). The rules of brainstorming apply here:

- Generate as many ideas as possible
- Do not allow criticism or debate
- Let the imagination soar
- Explore and combine ideas

Create Improvement Backlog Items

The team then cumulatively votes on up to three most viable solutions. These potential solutions are written as improvement [stories](#) and features, planned in the following PI Planning event. During that event, the RTE helps ensure that the relevant work needed to deliver the identified improvements is planned. This approach closes the loop, thus ensuring that action will be taken and that people and resources are dedicated as necessary to improve the current state.

Following this practice, problem-solving becomes routine and systematic, and team members and ART stakeholders can ensure that the train is solidly on its journey of relentless improvement.

Inspect and Adapt for Solution Trains

The above describes a rigorous approach to problem-solving in the context of a single ART. If the ART is part of a Solution Train, the I&A event will often include key stakeholders from the Solution Train. In larger value streams, however, an additional Solution Train I&A event may be required, following the same format.

Due to the number of people in a Solution Train, attendees at the large solution I&A event cannot include everyone, so stakeholders are selected that are best suited to address the problems. This subset of people consists of the Solution Train's primary stakeholders and representatives from the various ARTs and [Suppliers](#).

Learn More

- [1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.
- [2] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.
- [3] Derby, Esther, and Diana Larsen. *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf, 2009.

Last update: 22 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
 Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Future product development tasks can't be predetermined. Distribute planning and control to those who can understand and react to the end results.

—Michael Kennedy,
Product Development for the Lean Enterprise

There is no magic in SAFe . . . except

Adjust Cookie Settings

*maybe for PI
Planning.*

ÑAuthors

PI Planning

PI Planning is a cadence-based event for the entire ART that aligns teams and stakeholders to a shared mission and vision.

PI planning is essential to SAFe: If you are not doing it, you are not doing SAFe.

Introduction to PI Planning: A Quick Overview

Details

The Agile Manifesto states, “The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.” SAFe takes this to the next level with PI planning.

Where possible, everyone is face-to-face (virtually or physically), and these large-scale PI planning events now occur within many enterprises worldwide. They have clearly shown real financial ROI, not to mention the intangibles that happen when the team of Agile teams creates a social construct that is personally and collectively rewarding.

It may not always be practical for the entire [Agile Release Train \(ART\)](#) to collocate; however, in our current times, COVID-19 has created a situation where this isn’t an option. While physical face-to-face planning has benefits, the unwritten SAFe “rule” is that the people who do the work plan the work. Real-time, concurrent, virtual, face-to-face planning has now proven effective when physical presence is not possible. Indeed many ARTs have been flourishing in creating a hybrid situation where several teams join remotely, as shown below in Figure 1.

Find a Course:

The advanced topic article, [Distributed PI Planning with SAFe](#), provides additional guidance and considerations for successfully managing these scenarios.

Figure 1. Face-to-face PI planning. Remote teams are planning at the same time using video conferencing.

PI Planning has a standard agenda that includes a presentation of business context and [vision](#), followed by team planning breakouts—where the teams create their [Iteration](#) plans and objectives for the upcoming [PI](#). Facilitated by the [Release Train Engineer \(RTE\)](#), this event includes all members of the ART and occurs within the [Innovation and Planning \(IP\) Iteration](#). Holding the event during the IP iteration avoids affecting the scheduling or capacity of other iterations in the PI. PI Planning takes two days, although the ART can extend this timebox to accommodate planning across multiple time zones.

Business Benefits of PI Planning

PI planning delivers many business benefits, including:

- Establishing face-to-face communication among all team members and stakeholders
- Building the social network the ART depends upon
- Aligning development to business goals with the business context, vision, and [Team and ART PI objectives](#)
- Identifying dependencies and fostering cross-team and cross-ART collaboration
- Providing the opportunity for just the right amount of architecture and [Lean User Experience \(UX\)](#) guidance
- Matching demand to capacity and eliminating excess Work in Process (WIP)
- Fast decision-making

Inputs and Outputs of PI Planning

Inputs to PI planning include:

- Business context (see “[Content readiness](#)” below)

Roadmap and vision

Highest priority **Features** of the ART Backlog

A successful PI planning event delivers two primary outputs:

Committed PI objectives □ Each team creates a set of **SMART** objectives with the business value assigned by the Business Owners.

ART planning board □ Highlighting the new feature delivery dates, feature dependencies among teams, and relevant milestones

Preparation

PI planning is a significant event that requires preparation, coordination, and communication. It is facilitated by the RTE and event attendees, including **Business Owners**, **Product Management**, **Agile Teams**, **System** and **Solution Architects**, the **System Team**, and other stakeholders. The RTE must schedule all PI planning in advance to be well prepared. The active participation of Business Owners in this event provides an essential **Guardrail** on budgetary spending.

For the event to be successful, preparation is required in three major areas:

1. **Organizational readiness**
2. **Content readiness**
3. **Logistics readiness**

The following sections describe these three areas.

Organizational Readiness

Before PI planning, there must be strategy alignment among participants, stakeholders, and Business Owners. Critical roles are assigned. To address this in advance, however, event organizers must consider the following:

Planning scope and context □ Is the planning process's scope (product, system, technology domain) understood? Do we know which teams need to plan together?

Business alignment □ Is there reasonable agreement on priorities among the Business Owners?

Agile teams □ Do we have Agile teams? Are there dedicated team members and an identified **Scrum Master/Team Coach** and **Product Owner** for each team?

Content Readiness

It's equally important to have a clear vision and context so that the right stakeholders can participate. Therefore, the PI planning must include the following:

Executive briefing ☰ A briefing that defines the current business context

Product vision briefing(s) ☰ Briefings prepared by Product Management, including the top 10 features in the [ART Backlog](#)

Architecture vision briefing ☰ A presentation made by the CTO, [Enterprise Architect](#), or System Architect to communicate new [Enablers](#), features, and [Nonfunctional Requirements](#) (NFRs)

Logistics Readiness

Preparing an event to support a large number of attendees isn't trivial. This prep can include securing and preparing the space for physically collocated planning. For remote attendees or a fully distributed PI Planning, this also includes investment in the necessary technical infrastructure. Considerations include:

Locations ☰ Each location where planning takes place needs preparation in advance.

Technology and tooling ☰ Real-time access to information and tooling to support distributed planning or remote attendees

Communication channels ☰ Primary and secondary audio, video, and presentation channels must be available

Standard Agenda

The event follows an agenda similar to Figure 2. Descriptions of each item follow. For guidance on adapting this agenda to support planning across multiple time zones, refer to the advanced topic article, [Distributed PI Planning with SAFe](#).

Figure 2. Standard two-day PI planning agenda

Figure 2. Standard two-day PI planning agenda

Day 1 Agenda

Business context ⚡ A Business Owner or senior executive describes the current state of the business, shares the [Portfolio Vision](#), and presents a perspective on how effectively existing solutions address current customer needs.

Product/solution vision ⚡ Product Management presents the current vision (typically represented by the top ten or so upcoming features). They highlight changes from the previous PI planning event and any relevant milestones.

Architecture vision and development practices ⚡ The System Architect presents the architecture vision. Also, a senior development manager may introduce Agile-supportive changes to development practices, such as test automation, [DevOps](#), [Continuous Integration](#), and [Continuous Deployment](#), which the teams will adopt in the upcoming PI.

Planning context and lunch ⚡ The RTE presents the planning process and expected outcomes.

Team breakouts #1 ⚡ In the breakout, teams estimate their capacity for each [Iteration](#) and identify the backlog items they will likely need to realize the features. Each team creates draft plans, visible to all, iteration by iteration.

During this process, teams identify risks and dependencies and draft their initial team PI objectives. The PI objectives typically include “uncommitted objectives,” which are goals built into the plan (for example, stories that have been defined and included for these objectives) but are not committed to by the team because of too many unknowns or risks. Uncommitted objectives are *not* extra things to do in case there is time. Instead, they increase the reliability of the plan and give management an early warning of any objectives that the ART may not be able to deliver. The teams also add the features and associated dependencies to the ART Planning Board, as shown in Figure 3.

Figure 3. ART planning board showing features and dependencies

Figure 3. ART planning board showing features and dependencies

Draft plan review During the tightly timeboxed draft plan review, teams present key planning outputs, which include capacity and load, draft PI objectives, potential risks, and dependencies. Business Owners, Product Management, and other teams and stakeholders review and provide input.

Management review and problem-solving Draft plans likely present challenges

like scope, people and resource constraints, and dependencies. During the problem-solving meeting, management may negotiate scope changes and resolve other problems by agreeing to various planning adjustments. The RTE facilitates and keeps the primary stakeholders together for as long as necessary to make the decisions needed to reach achievable objectives.

Solution Trains often hold an additional management review and problem-solving workshop after the first day of planning to address cross-ART issues. Alternatively, the RTEs of the involved trains may talk with each other to discuss the problems for the ART's specific management review and problem-solving meeting. The **Solution Train Engineer (STE)** helps facilitate and resolve issues across the ARTs.

Day 2 Agenda

Planning adjustments □ The next day, the event begins with management presenting changes to the planning scope, people, and resources.

Team breakouts #2 □ Teams continue planning and making the appropriate adjustments. They finalize their objectives for the PI, to which the Business Owners assign business value, as shown in Figure 4.

Figure 4. A team's PI objectives sheet with assigned business value

Final plan review and lunch □ All teams present their plans to the group during this session. At the end of each team's time slot, the team states its risks and impediments and provides the risks to the RTE for use later in the ROAMing exercise.

The team then asks the Business Owners if the plan is acceptable. If the plan is accepted, the team brings their team PI objective sheet to the front of the room so everyone can see the aggregate objectives unfold in real-time. If the Business Owners have concerns, teams can adjust the plan to address the identified issues. The team then presents its revised plan.

ART PI Risks During planning, teams have identified risks and impediments that could impact their ability to meet their objectives. These are resolved in a broader management context before the whole train. One by one, the risks are discussed and addressed with honesty and transparency and then grouped into one of the following categories:

Resolved The teams agree that the risk is no longer a concern

Owned Someone on the train owns the risk since it cannot be addressed during PI planning

Accepted Some items are simply facts or potential problems that must be understood and accepted

Mitigated Teams identify a plan to reduce the impact of the risk

Confidence vote Once ART PI Risks have been addressed, teams vote on their confidence in meeting their team PI objectives

Each team conducts a vote using their fingers (fist of five) or a digital tool for remote events. If the average is three fingers or above, then management should accept the commitment. If it's less than three, the team reworks its plan. Anyone voting two fingers or fewer should be allowed to voice their concerns. These concerns might add to the risk list, require replanning, or provide information. Once each team has voted, it's repeated for the entire ART, with everyone expressing their confidence in the collective plan, as illustrated in Figure 5.

Figure 5. Confidence vote for an ART

Plan rework D If necessary, teams adjust their objectives until they have high confidence. This additional planning is one occasion where alignment and commitment are valued more highly than adhering to a timebox.

Planning retrospective and moving forward D Finally, the RTE leads a brief retrospective for the PI planning event to capture what went well, what didn't, and what to do better next time, as shown in Figure 6.

Figure 6. PI planning retrospective

Next steps Typically, a discussion about the next steps, along with final instructions to the teams, follows, including:

- Cleaning up the rooms used for planning (if applicable)
- Entering the team PI objectives and stories in Agile lifecycle management (ALM) tooling
- Reviewing team and ART events calendars
- Determining [Iteration Planning](#) and Team Sync locations and timing

After the planning event, the RTE and other ART stakeholders summarize the individual team PI objectives into a set of ART PI objectives (Figure 7) and use this to communicate externally and track progress toward the goals.

Product Management uses the ART PI objectives to refine the roadmap, improving the forecast for the following two PIs.

The ART Planning board is often used during the [Coach Sync](#) to track dependencies. It may or may not be maintained (manually) after planning is complete. A digital tool for managing dependencies facilitates their follow-up.

Teams leave the PI planning event with a prepopulated iteration backlog for the upcoming PI. They take their team's PI objectives, iteration plans, and risks to their regular work area. ART risks remain with the RTE, which ensures that the people responsible for owning or mitigating a risk have captured the information and are actively managing the risk.

Most importantly, the ART executes the PI, tracking progress and adjusting as necessary as new knowledge emerges. Execution of the PI begins with all the teams conducting planning for the first iteration, using their PI plans as a starting point. It offers fresh input for the iteration planning processes that follow. Since the iteration plans created during PI Planning did not consider detailed story-level acceptance criteria, the team will likely adjust the first and subsequent iteration plans.

Figure 7. ART PI objectives

Figure 7. ART PI objectives

Solution Train PI Planning

This article focuses on the planning activities of a single ART. However, large [Value Streams](#) may contain multiple ARTs and suppliers. In this case, the Solution Train provides coordination using [Pre-Plan](#) and [Coordinate and Deliver](#) activities.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[2] Kennedy, Michael. *Product Development for the Lean Enterprise*. Oaklea

Last update: March 19, 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)[Posters & Graphics](#)[Watch and download
SAFe videos and
presentations](#)[Blog](#)

TRAINING

[Course Calendar](#)[About Certification](#)[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)[Partner Directory](#)

GET SOCIAL

[Twitter](#)[Linkedin](#)[YouTube](#)CONTENT &
TRADEMARKS[FAQs on how to use
SAFe content and
trademarks](#)[Permissions Form](#)[Usage and Permissions](#)SCALED AGILE,
INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS
HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



There's innovation in Linux. There are some really good technical features that I'm proud of. There are capabilities in Linux that aren't in other operating systems.

Linus Torvalds,
creator of Linux

Adjust Cookie Settings

Features and Capabilities

A Feature represents solution functionality that delivers business value, fulfills a stakeholder need, and is sized to be delivered by an Agile Release Train within a PI.

Each feature includes a benefit hypothesis and acceptance criteria and is sized or split as necessary to be delivered by a single Agile Release Train (ART) in a PI.

A Capability represents large solution functionality whose implementation often spans multiple ARTs and is sized to be delivered within a PI.

Features also lend themselves to the [Lean UX](#) process model, which includes a definition of the Minimum Marketable Feature (MMF), a benefit hypothesis, and acceptance criteria. The MMF helps limit the scope and investment, enhances agility, and provides fast feedback. Capabilities behave the same way as features. However, they are at a higher level of abstraction and support the definition and development of large [Solutions](#).

Details

Features and capabilities are critical to defining, planning, and implementing [Solution](#) value. Figure 1 provides a broader context for these work items:

Figure 1. Features and Capabilities in the SAFe context

Figure 1. Features and Capabilities in the SAFe context

Figure 1 shows that solutions are developed using features. Each reflects a service provided by the system that fulfills some important stakeholder needs. They are maintained in the [ART Backlog](#) and sized to fit in a [PI](#) so that each delivers new value. Features can originate from either the [Agile Release Train \(ART\)](#)'s local context or from splitting [Epics](#) or capabilities.

The [ART and Solution Train Kanban](#) systems support the flow of features and capabilities, where they progress through the funnel, analyzing, backlog, implementing, validating, deploying, and releasing states. This process provides reasoned economic analysis, technical impact, and strategy for incremental implementation.

[Product Management](#) and [System Architect](#) define the features and enablers, respectively. [Nonfunctional Requirements](#) (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. NFRs serve as constraints or restrictions on the system's design across the different

backlogs. Features are prioritized using [Weighted Shortest Job First \(WSJF\)](#) and are planned and reviewed at PI boundaries. They are split into [Stories](#) and are implemented, integrated, tested, and demonstrated as the functionality becomes available.

Discovering and Describing Features

[Design Thinking](#) takes a [Customer-Centric](#) approach to creating desirable and sustainable products. Design thinking tools, including personas, empathy maps, and customer journey maps, provide empathy towards and a deeper understanding of customers and users. Together, they offer a rich context to better understand features and their potential benefits.

Features are defined using a features and benefits format:

Feature ☐ A short phrase giving a name and context

Benefit hypothesis ☐ The proposed measurable benefit to the end user or business

Avoid defining features with the “[User story voice](#)” format designed to support one user role; features typically provide functionality for multiple user roles. Furthermore, using the same method to describe user stories and features may cause confusion.

Figure 2 illustrates an example set of features with benefits hypotheses:

Figure 2. Features and benefits hypotheses

Figure 2. Features and benefits hypotheses

Creating and Managing Features

In collaboration with [Product Owners](#) and other key stakeholders, Product Managers define features in the local context of an ART. Some arise as a result of splitting epics.

System Architects typically create enabler features. The ART backlog is used to maintain enablers alongside business features. Enablers pave the [Architectural Runway](#) and support exploration or provide the infrastructure needed to develop, test, and integrate the solution.

Like business features, enabler features may originate from epics or emerge locally at the ART level. Enablers that make it through the [Kanban](#) system will be subject to capacity allocation in the ART backlog to ensure enough emphasis on furthering the solution and extending the architectural runway. At each PI boundary, the percentage of resources allocated to new features (or capabilities) versus enablers

is estimated to guide the train.

Prioritizing Features

The WSJF prioritization model is used to sequence jobs (e.g., features, capabilities) based on the economics of product development flow. Since implementing the *right jobs* in the *right sequence* produces the maximum economic benefit¹ it is hard to overstate the importance of this critical process.

Product and Solution Management use WSJF to prioritize features, while System and Solution Architects also use WSJF to prioritize enabler features. Since business features and enabler features exist in the same backlog, Product and Solution Management must work collaboratively with System and Solution Architects to reconcile differences in priorities. Aligning priorities to [Strategic Themes](#) and capacity allocation are two approaches to creating alignment and balance in the backlog.

Estimating Features

Feature estimation supports forecasting value delivery, applying WSJF prioritization, and sizing epics by splitting them into features and summing their estimates. Feature estimation usually occurs in the analysis state of the ART Kanban. It relies on normalized estimation techniques, similar to the methods used by Agile teams (see the [Iteration Planning](#) article for more detail). During analysis, select subject matter experts from the ART engage in exploration activities and preliminary sizing.

Accepting Features

Feature acceptance criteria determine whether the implementation is correct and delivers the business benefits. Figure 3 provides an example:

Figure 3. Feature with acceptance criteria

Figure 3. Feature with acceptance criteria

Acceptance criteria mitigate implementation risk and enable early validation of the benefit hypothesis by creating alignment between product management, stakeholders, and developers. Acceptance criteria can also be used as the source of stories. As with stories, acceptance criteria are often transformed into acceptance tests with [Behavior-Driven Development](#) (BDD).

Product Management is responsible for accepting the features. They use acceptance criteria to determine whether the functionality is implemented correctly and whether nonfunctional requirements are met.

Capabilities

Most of this article is devoted to describing the definition and implementation of features, as they are the most common description of system behavior. Capabilities exhibit the same characteristics and practices as features. For example, they:

- Are described using a phrase and benefit hypothesis
- Are sized to fit within a PI; however, they often take multiple ARTs to implement
- Are reasoned about and approved using the Solution Train Kanban. The [Solution Train Backlog](#) holds approved capabilities
- Have associated enablers to describe and bring visibility to all the technical work necessary to support the efficient development and delivery of business capabilities
- Are accepted by [Solution Managers](#), who use the acceptance criteria to determine whether the functionality is fit for purpose

Capabilities may originate in the local context of the solution or occur as a result of

splitting portfolio epics that may cut across more than one [Value Stream](#). Another potential source of capabilities is the [Solution Context](#), where some aspects of the environment may require additional solution functionality.

Splitting Features and Capabilities

Capabilities must be decomposed into features to be implemented. They, in turn, are split into stories consumable by teams within an iteration. SAFe provides ten patterns for splitting work, as described in Leffingwell [1], chapter 6.

1. Workflow steps
2. Business rule variations
3. Major effort
4. Simple/complex
5. Variations in data
6. Data methods
7. Deferring system qualities
8. Operations
9. Use-case scenarios
10. Breaking out a spike

Figure 4 illustrates splitting a capability into features.

Figure 4. A capability split into features

Figure 4. A capability split into features

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 14 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile

Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



It is a misuse of our power to take responsibility for solving problems that belong to others.

~Peter Block

Release Train Engineer

The Release Train Engineer (RTE) is a servant leader and ART coach who facilitates ART events and processes, and supports teams in delivering value.

They communicate with stakeholders, escalate impediments, help manage risk, and drive relentless improvement.

Although [Agile Release Trains \(ARTs\)](#) are composed of self-organizing and self-managing teams, these trains cannot drive or steer themselves. That is the responsibility of the RTE, who facilitates most effectively as a servant leader. They have a solid grasp of how to scale Lean and Agile practices and understand the unique opportunities and challenges of aligning and facilitating an ART (team-of-Agile Teams).

Details

The RTE facilitates ART practices and PI execution. They escalate impediments, manage risk, and help ensure value delivery and relentless improvement. RTEs also often participate in the Lean-Agile transformation, coaching leaders, teams, and [Scrum Master/Team Coaches](#) in the new mindset and processes. They also help adapt SAFe to the organization's needs, standardizing and documenting practices.

Responsibilities

The RTE is focused primarily on the five following types of activities illustrated in Figure 1. The following sections describe each responsibility.

Figure 1. Release Train Engineer primary responsibilities

Figure 1. Release Train Engineer primary responsibilities

Facilitating PI Planning

PI Planning is a cadence-based, face-to-face event that serves as the heartbeat of the ART, aligning all the teams on the ART to a shared mission and [Vision](#). RTEs play a vital role in this critical event and typically:

Help the ART Prepare for PI Planning ☰ The RTE helps ensure planning readiness in three primary areas: 1) Strategic alignment and organizational readiness for planning (planning scope and context), 2) leadership and team preparedness for the event (content readiness), and 3) manage the logistics for the event (facility or technology and tooling for remote events). RTEs also help the ART prepare by fostering a [Continuous Exploration](#) process that drives the synthesis of a vision, a [Roadmap](#), and [Backlogs](#).

Facilitate the PI planning event ☰ An effective RTE is critical to a successful event, and they typically do the following activities to help facilitate PI planning:

On day 1 of planning, the RTE opens the event and reviews the purpose,

agenda, working agreements, planning rules, expectations, and other logistics. They introduce speakers who discuss the business context, product vision (and top 10 features), architectural vision, and development practices and present the planning context. They run the Coach Sync (during team breakout #1) and manage the draft plan reviews and management review problem-solving meetings.

On day 2 of planning, the RTE facilitates planning adjustments, team breakout #2 and Coach Syncs, final plan reviews, addresses ART PI risks, holds a confidence vote, plans rework (if needed), and ends the event with the planning retrospective and moving forward instructions.

Supporting PI Execution

RTEs have significant accountability for the successful execution of the PI. They typically have the following responsibilities during the PI:

Track progress RTEs assist in tracking the execution of [Features](#) in the ART Kanban, ensuring the ART meets its definition of done. RTEs coordinate impediment removal and escalate and address problems that the teams cannot resolve independently. Moreover, they use working solutions in the System Demo as the primary measure of progress.

Facilitate periodic synchronization events These events include the ART sync, [System Demos](#), and PI system demo. They help the ART manage PI risks and dependencies using the ROAM technique and ART planning board.

Support ART backlog refinement RTEs collaborate with [Product](#) and [Solution Management](#), Business Owners, [Product Owners](#), and other stakeholders to help ensure the backlog aligns with strategy.

Promote DevOps and Continuous Delivery RTEs promote [DevOps](#), and the [Continuous Delivery Pipeline](#), including [Built-in Quality](#) and the [Lean User Experience \(UX\)](#) innovation cycle. Moreover, they help coordinate releases and plan additional activities and milestones needed to deliver the solution, so it meets the definition of done (DoD).

Assist Business Owners RTEs support economic decision-making for epics, facilitating feature estimation by [Agile Teams](#). They understand and operate within [Lean Budgets](#), ensuring adherence to [Guardrails](#).

Coordinate planning efforts with other ARTs RTEs establish and communicate the annual calendars for [Iterations](#) and [PIs](#) and schedule any pre-planning activities.

Coaching the ART

RTEs are the ART's chief Scrum Master/Team Coach. In this role, they generally have the following types of responsibilities:

Coach with powerful questions RTEs do not have all the answers. Instead, they ask *powerful questions* to uncover what's essential, then guide others to tap into their knowledge and expertise. Some examples of powerful questions include:

- What brings us to this inquiry?
- What other possibilities or options exist?
- What is it we're not seeing?
- What do we need to reach a deeper level of understanding?
- If success was guaranteed, what actions would you take?

Coach SM/TCs in ART and Team events and practices This may include ART events such as PI planning, System Demos and [Inspect and Adapt](#), and team events, iteration planning, review, retrospective, and backlog refinement.

Coach ART roles RTEs coach Business Owners, System Architects, and Product Management and encourage collaboration between teams and [Systems Architects](#). In addition, they foster Lean-Agile practices and mindsets for Agile Teams and the ART.

Optimizing Flow

Agile Teams and ARTs strive to achieve a state of continuous flow, enabling new features to move quickly from concept to cash. [Principle #6, Make value flow without interruption](#), describes how to accelerate flow. RTEs are typically responsible for the following flow activities:

Establish pull systems to optimize the flow of value RTEs use various tools, such as the ART Kanban and other information radiators, to ensure a smooth flow of value.

Establish ART flow measures These metrics include the five flow measures (see [Measure and Grow](#)) and *flow predictability*, which identifies how predictable Agile Teams and trains deliver business value against their planned objectives.

Improve the flow of value RTEs help improve the flow of value by assessing and improving the practices associated with DevOps and the Continuous Delivery Pipeline. Moreover, they coach the train to apply the eight flow accelerators described in the [Team Flow](#) and [ART Flow](#) articles.

Facilitate value stream mapping RTEs help the ART define the development value stream's steps, identifying handoffs, bottlenecks, and delays. This mapping highlights the total lead time needed to fulfill a request while spotlighting areas for improvement.

Reduce or eliminate cross-team dependencies RTEs review patterns from the ART planning board and consider how to improve the organizational design of the ART and teams by applying [Team Topologies](#).

Improving Relentlessly

The relentless pursuit of perfection has always been one of the core tenets of Lean. It's also one of SAFe's four core values. While unattainable, striving for perfection leads to continuous improvements to products and services. The RTE typically facilitates the following types of improvement activities:

Drive relentless improvement □ RTEs foster the pursuit of perfection via the [Inspect and Adapt](#) problem-solving workshop. They support just-in-time improvement throughout the PI, leveraging the Coach and PO syncs, [Communities of Practice](#), and promote the use of engineering and [Built-In Quality](#) practices.

Leverage the SAFe core competency self-assessments □ RTEs help teams and trains improve on the technical and business practices needed to achieve the larger aim of the portfolio.

Collaborate with the Value Management Office (VMO) and LACE □ RTEs help focus the ART on delivering value and operational excellence (see the [Lean Portfolio Management article](#)).

RTEs Are Servant Leaders

While new RTEs typically have the organizational skills to perform their roles, they may need to learn and adopt [Lean-Agile Mindsets](#). They may need to transition from directing and managing activities to acting as a servant leader. Servant leadership is a philosophy that implies a comprehensive view of the quality of people, work, and community spirit [1]. The focus is on providing the support needed by the teams and ARTs to be self-organizing and self-managing. Characteristic servant leader actions include:

- Listen to and support teams in problem identification and decision-making
- Create an environment of mutual influence
- Understand and empathize with others
- Encourage and support the personal development of each individual and the development of teams
- Think beyond day-to-day activities and [apply systems thinking](#)
- Support the teams' commitments
- Be open and appreciate openness in others

Learn More

[1] Servant Leadership. http://en.wikipedia.org/wiki/Servant_leadership

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[3] Trompenaars, Fons, and Ed Voerman. *Servant-Leadership Across Cultures: Harnessing the Strengths of the World's Most Powerful Management Philosophy*. McGraw-Hill, 2009.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



In order for you to keep up with customer demand, you need to create a deployment pipeline. You need to get everything in version control. You need to automate the entire environment creation process. You need a deployment pipeline where you can create test and production

Adjust Cookie Settings

*environments, and
then deploy code
into them, entirely
on demand.*

ÑErik to
Grasshopper, *The
Phoenix Project*

Continuous Deployment

Continuous Deployment (CD) is an aspect of the Continuous Delivery Pipeline that automates the migration of new functionality from a staging environment to production, where it is made available for release.

CD is the third aspect in the four-part [Continuous Delivery Pipeline](#) (CDP) of [Continuous Exploration](#) (CE), [Continuous Integration](#) (CI), Continuous Deployment (CD), and [Release on Demand](#) (Figure 1).

Figure 1. Continuous deployment in the context of the CDP

Figure 1. Continuous deployment in the context of the CDP

Features must be available and verified in production **before** the business needs them to support [Release on Demand](#). Therefore, it's optimal to separate the deployment process from releasing, enabling changes to move into production without affecting the behavior of the current system. Continuous deployment allows teams to deploy small, incremental changes to production continually.

The capability to continuously deploy is critical for releasing on demand. In turn, it allows Agile Release Train (ARTs) to respond to market opportunities with the highest possible value in the shortest sustainable lead time, permitting customers to consume new functionality when they are ready.

Details

Traditional development practices treat deployment and release as the same activity. In this model, changes deployed to production are immediately available to users. Continuous deployment, however, separates the deployment and release

processes. This practice fosters design thinking and fast value flow by:

Targeting functionality to specific customers **D** Enables the organization to target customers with particular functionality, allowing the organization to assess the impact of changes before deploying functionality to all customers.

Promoting experimentation, such as A/B Testing **D** Design thinking practices, such as A/B testing, require the ability to present different functionality to distinct target users, gathering feedback that helps create the optimal user experience.

Promoting small batches **D** Automating the CDP (for example, tests, builds, deploys) makes deploying in small batches economically feasible.

Releasing on business needs **D** ARTs tend to release less frequently when the deployment process is complex and error-prone. Organizations that invest in automation and relentless process improvement can release faster and with lower risk, substantially increasing [Business Agility](#). For example, a release can be deployed in production ahead of a marketing campaign, giving the organization more flexibility in maximizing all aspects of value delivery.

To enable these capabilities, ARTs focus on reducing the transaction cost and risk of moving changes to production by automating all aspects of continuous deployment. Ensuring the deployment process is a repeatable, predictable activity without significant incidents helps teams achieve continuous deployment. Moreover, improving deployment to make value flow without interruption [Principle #6](#)) is critical for achieving business agility. See the [ART Flow](#) article for more information.

The Four Activities of Continuous Deployment

SAFe describes four activities of Continuous Deployment, as illustrated in Figure 2.

1. **Deploy** **D** the practices necessary to deploy a solution to a production environment
2. **Verify** **D** the practices needed to ensure solution changes operate in production as intended before releasing them to customers
3. **Monitor** **D** the practices to monitor and report on any issues that may arise in production
4. **Respond** **D** the practices to address any problems rapidly which may occur during deployment

Figure 2. Four activities of continuous deployment

Figure 2. Four activities of continuous deployment

Deploy

Deployment is the migration of changes into a production environment. In the CDP, deploying changes is done continuously. Partial functionality can be implemented incrementally into production.

Suppose a user story map has 27 stories to implement a new workflow. Traditional legacy practices would likely result in all 27 stories deployed in one large batch. Instead, individual stories can be deployed in “dark mode” with continuous deployment and feature toggles. When the ART has deployed the full features set, the business can choose when to release them to users.

Ideally, the deployment pipeline triggers the deployment process automatically following a successful build, integration, and validation. This approach makes the workflow a fully-automated one-click process, from code-commit to production-deploy. Highly sophisticated enterprises can reliably deploy anytime, even during peak periods. This approach eliminates the need to work weekends, nights, or

other off-hours to deploy.

Several practices contribute to the ability to deploy:

Dark launches ▷ the ability to deploy new functionality to a production environment without releasing it to end users

Feature toggles ▷ a technique to facilitate dark launches by implementing toggles in the code, which enables switching between old and new functionality

Deployment automation ▷ the ability to deploy a tested solution automatically from check-in to production

Selective deployment ▷ the ability to deploy to specific production environments and not others based on criteria such as geography, market segment, and more

Self-service deployment ▷ when automated deployment is partially implemented, self-service allows a single command to move solutions from staging to production

Version control ▷ maintaining environments under version control enables fast deployment and recovery

Blue/green deployment ▷ a technique that permits on-demand switching between staging and production environments

Verify

Deployments must be verified for functional integrity and robustness before releasing to end users. These two processes almost happen simultaneously when tightly coupled, making recovery decisions a primary concern. However, when they are separated, there's room to test new functionality extensively in production *before* approving it for release. After migration to production, solutions undergo a final round of testing. Typically, this requires a smoke test, light user acceptance testing, and a stress and performance test, which must occur in a production setting. This verification provides the necessary sanity check that tests the behavior of the solution in its actual production [Solution Context](#).

[Continuous Integration](#) reasonably ensures that the solution will behave as expected in production; however, surprises do occur. When verification reveals critical defects, deployments must either be rolled back or fixed quickly to prevent them from harming the production environment or disrupting the business flow.

Four practices help drive verification after deployment:

Production testing ▷ testing solutions in production using feature toggles or dark launches

Test automation ▷ the ability to automate tests and run them rapidly and repeatedly

Test data management ▷ managing test data in version control to ensure consistency in automated testing

Testing nonfunctional requirements (NFRs) ▷ teams also test system attributes

such as security, reliability, performance, scalability, and usability to ensure NFRs meet quality standards

Monitor

Verifying that deployed features didn't break on their way into production is an essential pre-release quality check. However, teams must also ensure they can measure a feature's performance and value over its lifespan. The insights that drive this critical feedback loop primarily come from robust monitoring capabilities, which must be in place before release.

Effective monitoring requires that full-stack telemetry is active for all features deployed through the CDP. This telemetry allows teams to verify system performance, end-user behavior, incidents, and business value rapidly and accurately in production. The data collected provides tracking and monitoring of each feature, increasing the fidelity of analysis of the business value delivered and increasing responsiveness to production issues.

While teams cannot collect some business-value metrics until release, they need to know how to obtain the measures before the release decision occurs. Some practices which help support this include:

Full-stack telemetry ▷ the ability to monitor for problems across the entire stack that a system covers

Visual displays ▷ tools that display automated measurements

Federated monitoring ▷ consolidated monitoring across applications in the solution that creates a holistic view of problems and performance

The [Measure & Grow](#) article provides some guidance on the types of metrics that monitoring requires.

Respond

The ability to respond to and recover from unforeseen production issues is critical to supporting continuous deployment and streamlining the CDP. The reasons are obvious:

Production issues directly affect customers and end users, so the value of deployed solutions can quickly erode when problems occur.

Production issues causes rework/fixes, patches, redevelopment, retesting, redeployment, etc. That disrupts the normal flow of value through the pipeline.

Since production issues can harm delivery efficiency and lower value, teams need the capability to detect problems proactively and recover quickly. As measured by Mean Time to Restore (MTTR), fast recovery is among the most reliable leading

indicators of high [DevOps](#) maturity [5]. Recovery is also one of the five elements of SAFe's [CALMR](#) approach to DevOps.

The goal of responding and recovering is to identify potential issues *before* they turn into incidents and to prevent them from affecting business operations. This capability requires detecting difficulties internally before end users discover them, quickly identifying root causes, and restoring services with well-rehearsed procedures. In contrast, making hasty, reactive changes directly to production systems—just to keep the lights on—invites source code and configuration differences between environments, unverified changes, and long-term risk.

Several practices support the ability to respond and recover from production issues:

Proactive detection △ a technique for proactively creating faults in the solution to identify potential problems and situations before they occur. For example, Chaos Monkey [1], developed by Netflix, is an open-source tool that randomly terminates instances in production to ensure that engineers implement their services to be resilient to instance failures.

Cross-team collaboration △ a mindset of cooperation across the [Value Stream](#) to identify and solve problems as they arise

Session replay △ the ability to replay end-user sessions to research incidents and identify problems

Rollback and fix forward △ the ability to both rollback a solution quickly to a previous environment or to fix a problem quickly through the pipeline without the need to rollback

Immutable infrastructure △ This concept refers to never amending servers or Virtual Machines (VMs) after deployment. Instead, a new server is built from an image with the appropriate changes if something needs updating.

Version control △ environments should be maintained under version control to rollback quickly

After teams have demonstrated that features have been deployed successfully to production and have the necessary monitoring and recovery capabilities to track and manage ongoing value, they have completed the *continuous deployment* stage of the CDP. In turn, this gives the enterprise the ability to release whenever warranted.

Enabling Continuous Deployment with DevOps

Continuous deployment involves critical operation activities frequently associated with the “Ops” in DevOps. They focus on deploying solutions to production environments, verifying their functional integrity, and ensuring effective monitoring

and post-release support.

Figure 3 illustrates that SAFe's CALMR approach to DevOps (center) and several practice domains (inner rings) enables continuous deployment. Each of the four activities (in green) is a collaborative effort that draws upon DevOps expertise from multiple disciplines to maximize delivery speed and quality.

Figure 3. DevOps enables continuous deployment

Figure 3. DevOps enables continuous deployment

For instance, *deploying* solutions in the CDP involves using tools that automate the provisioning of production infrastructure, deploy solution binaries to select targets, verify production functionality, capture runtime telemetry, and proactively alert on issues. DevOps practices and tools streamline these capabilities, allowing solutions to be deployed and fully prepared for on-demand release in minutes.

All four continuous deployment activities are enabled by DevOps, though with different combinations of technical practices and tooling. See the [DevOps](#) article

series for more guidance on DevOps and how it facilitates the CDP.

Learn More

[1] <https://netflix.github.io/chaosmonkey/>

[2] Kim, Gene, et al. *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. IT Revolution Press, 2013.

[3] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

[4] Humble, Jez, and David Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley, 2010.

[5] Gregory, Janet, and Lisa Crispin. *More Agile Testing: Learning Journeys for the Whole Team*. Addison-Wesley Signature Series (Cohn). Pearson Education, 2014.

[6] State of DevOps Report. <https://puppet.com/resources/whitepaper/state-of-devops-report>

Last update: 9 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

[Download SAFe
Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

INC

CONTACT US

5400 Airport Blvd., Suite
300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



The more alignment you have, the more autonomy you can grant. The one enables the other.

—Stephen Bungay,
author and strategy
consultant

Agile Release Train

The Agile Release Train (ART) is a long-lived team of Agile teams that incrementally develops, delivers, and often operates one or more solutions in a value stream.

[Adjust Cookie Settings](#)

Details

ARTs are teams of [Agile Teams](#) that align to a shared business and technology mission. Each is a virtual organization (typically 50 – 125 people) that plans, commits, develops, and deploys together. ARTs are organized around the enterprise’s significant [Development Value Streams](#) and exist solely to realize the promise of that value by building and delivering [Solutions](#) that benefit the [Customer](#).

ARTs are cross-functional and have all the capabilities needed to define, build, validate, release, and, where applicable, operate solutions.

These capabilities allow the ART to deliver a *continuous flow of value*, as shown in Figure 1.

Figure 1. A long-lived Agile Release Train

Figure 1. A long-lived Agile Release Train

ART Characteristics

Organized Around Value

As virtual organizations, ARTs have all the people needed to define, deliver, and operate the solution, eliminating the functional silos that may exist, as shown in Figure 2.

Figure 2. Traditional functional organization

Figure 2. Traditional functional organization

In a “functional” organization, developers work with developers; testers collaborate with other testers; architects and systems engineers work with each other, and operations work by themselves. Although there are reasons why organizations have evolved this way, the structure slows the flow of value, as it must cross all the silos. The daily involvement of managers is necessary to move the work across silos. As a result, progress is slow, and handoffs and delays rule the day.

Instead, the ART applies systems thinking (SAFe Principle #2) and organizes around value (SAFe Principle #10) to build an optimized cross-functional organization. This

facilitates the flow of value from ideation through deployment and release into operations, as Figure 3 illustrates.

Figure 3. Agile Release Trains are fully cross-functional

Figure 3. Agile Release Trains are fully cross-functional

Together, this fully cross-functional organization—whether physical (direct organizational reporting) or virtual (line of reporting is unchanged)—has everyone and everything it needs to define, deliver, and operate solutions. It is self-organizing and self-managing. This creates a far leaner organization, where traditional daily task and project management is no longer required. Value flows more quickly with less overhead.

To simplify the job of finding the optimum structure of ARTs within the organization and Agile teams within ARTs, SAFe recommends team topologies as described in the book *Team Topologies* [1]. SAFe recommends four ways to organize teams (Figure 4).

Stream-aligned teams are end customer-aligned and are capable of performing all the steps needed to build end-to-end customer value.

Complicated subsystem teams are organized around critical solution subsystems.

They focus on areas of high technical specialization, which limits the cognitive load on all the teams.

Platform teams provide application services and APIs for stream-aligned teams to be able to leverage common platform services.

Enabling teams provides tools, services, and short-term expertise to other teams.

Figure 4. Applying team topologies to Agile teams on an ART

Figure 4. Applying team topologies to Agile teams on an ART

Further guidance on organizing Agile teams can be found in the extended guidance article [Organizing Agile Teams and ARTs: Team Topologies at Scale](#).

Agile Teams Power the Train

ARTs include the Agile teams that define, build, and test features, as well as those that deploy, release, and operate the solution. SAFe Agile teams apply [SAFe Scrum](#) or [SAFe Team Kanban](#) or hybrids that suit their specific context. Each Agile team typically has ten or fewer dedicated individual contributors covering all the roles necessary to build a quality increment of value. Teams may be technology-focused

Ñdelivering software, hardware, and any combinationÑbusiness-focused, or a combination of both. Each Agile team has two specialty roles, the [Scrum Master / Team Coach](#) and the [Product Owner](#). And, of course, Agile teams within the ART are themselves cross-functional, as shown in Figure 5.

Figure 5. Agile teams are cross-functional

Figure 5. Agile teams are cross-functional

Aligned to a Common Cadence

ARTs also address one of the most common problems with traditional Agile development: teams working on the same solution operate independently and asynchronously. That makes it extremely difficult to routinely integrate the entire system. In other words, ÔThe teams are iterating, but the system isnÕt.Õ This increases the risk of late discovery of issues and problems, as shown in Figure 6.

Figure 6. Asynchronous Agile development

Figure 6. Asynchronous Agile development

Instead, the ART applies cadence and synchronization to assure that the system is iterating as a whole (Figure 7).

Figure 7. Aligned development: this system is iterating

Figure 7. Aligned development: this system is iterating

Cadence and synchronization ensure the focus remains on the evolution and objective assessment of the full system rather than its elements. The [System Demo](#), which occurs at the end of every [Iteration](#), provides objective evidence that the system is iterating. As Figure 7 illustrates, a [System Team](#) is often formed as an enabling group to help with infrastructure development and full system integration and validation. Over time, however, many of the centralized services provided by the System Team can be automated or absorbed by the teams themselves.

Enabled by Critical Roles

In addition to the Agile teams, the following roles aid the successful execution of the ART:

[Release Train Engineer \(RTE\)](#) is a servant leader who facilitates ART execution, impediment removal, risk and dependency management, and continuous improvement.

Product Management is largely responsible for what gets built, as defined by the **Vision**, **Roadmap**, and new **Features** in the **ART Backlog**. They work with customers, teams, and Product Owners to understand and communicate their needs and participate in solution validation.

System Architect is an individual or team that defines the system's overall architecture. They work at a level of abstraction above the teams and components and typically define **Nonfunctional Requirements (NFRs)**, major system elements, subsystems, and interfaces.

Business Owners are key stakeholders of the ART, with final responsibility for the business outcomes of the train.

Customers are the ultimate economic buyers or value users of the solution.

In addition to these critical ART roles, the following functions play an essential part in ART success:

System Teams typically assist in building and maintaining development, continuous integration, and test environments.

Shared Services are specialists necessary for the success of an ART but cannot be dedicated to a specific train. They often include data security, information architects, site reliability engineering (SRE), database administrators (DBAs), and many more.

Key ART roles rely on support from the teams. For example, the RTE depends on Scrum Master / Team Coaches for help with aspects of ART operations and improvement. Product Management relies on Product Owners to turn their product vision into reality. Architects collaborate with technology professionals on teams to devise viable architectures.

ART Responsibilities

The ultimate purpose of every ART is to deliver effective solutions to the customer. Essentially, ARTs are built for the sole purpose of establishing a fast flow of solution features. To achieve that, a train develops the solution iteratively, constantly engaging with the customer and adjusting the course of action towards an optimal solution.

Figure 8 shows the critical areas of responsibility of an ART that help achieve that objective:

Figure 8. ART responsibilities

Figure 8. ART responsibilities

Connecting with the Customer

Customers are the ultimate beneficiaries of the business solutions ARTs create and maintain. But connecting with the customer requires deliberate effort and a clear understanding of how to apply lean and agile practices in a unique ART context.

Apply customer centricity ▷ An ART routinely focuses on customer needs and opportunities to benefit the customer. **Customer Centricity** is a necessary mindset for the ART and its constituent teams. The ART works to increase and maintain customer empathy and continuously research better ways to solve customer problems.

Use design thinking ▷ A recurrent process of understanding the problem and designing the right solution. **Design Thinking** enables an ART to create desirable, feasible, and sustainable solutions. Paying close attention to user personas, journey mapping, and customer benefit analysis helps an ART discover new, valuable product capabilities. The use of lightweight prototypes validates customer value hypotheses quickly and keeps the ART on the right track.

Planning the Work

Planning crucial activities for an ART enables alignment across teams and stakeholders in terms of what and how to build within the next timebox. Alignment is one of the [Core Values](#) of SAFe, and ARTs, as a building block of a SAFe organization, have built-in means for achieving and sustaining alignment.

Align ART priorities with portfolio strategy

Every ART operates in a broader portfolio context and needs to align with the overall portfolio strategy. [Strategic Themes](#) orient the ARTs within a portfolio towards a common goal. However, achieving the alignment also requires an established process that involves: 1) regularly engaging with portfolio stakeholders at the ART level and 2) including ART representatives in portfolio interactions. Organizing this communication and interaction is easier around PI cadence. [Epic Owners](#) often serve as an important link between portfolio strategy and ART execution.

Prepare for PI Planning Stakeholders and teams need to prepare carefully for [PI Planning](#). Product Management and Business Owners develop the vision and agree on priorities for the next PI: teams take inventory of their remaining work, their attainable capacity, and any new effort that may emerge in the local context.

Plan the PI PI planning generates alignment within the ART. Teams create and agree on the [PI Objectives](#) that will guide them throughout the PI execution. Business Owners have an opportunity to share business and customer context with teams and, in turn, learn how the current technology and delivery capability can be employed to create optimal business value for the enterprise.

Delivering Value

ARTs develop solution features by applying a cadence that involves key activities to keep the train on the tracks. At certain points, an ART will release the newly created value to the customer.

Frequently integrate and test

A fast development rhythm requires frequent integration and testing. This helps uncover technology and implementation problems early and gives the teams enough time to respond to the findings. Without recurring integration and testing, an ART operates in excessive uncertainty and variability. [Built-in Quality](#) and [Team and Technical Agility](#) provide guidance on these practices.

Develop in short increments of value ☰ An ART implements the PI as a series of short increments, each representing a small batch of integrated, tested, and demonstrable value. The ART’s iteration cadence provides a natural pace to create these increments. Each helps the ART learn about potential implementation challenges, get customer feedback, and agree on a decision point with possible course corrections for the rest of the PI.

Regularly synchronize and make adjustments ☰ While executing the PI, an ART has multiple checkpoints in the form of an ART Sync, which includes a Coaches Sync and PO Sync (see the [PI](#) article for further description). These events increase visibility into the progress toward the current PI objectives and help the ART make timely adjustments.

Build a continuous delivery pipeline ☰ An effective Agile development process provides the means for ongoing exploration and integration of work. Additionally, the teams need to establish a continuous deployment process via building a [Continuous Delivery Pipeline](#) (CDP). This requires value stream mapping to identify the sources of excessive delay and variability. As a part of CDP, [Continuous Deployment](#) often involves purposeful system design that favors low coupling of capabilities, which enables the teams to deploy value independent of each other.

Establish release governance process ☰ Each ART establishes a governance process suitable for its release cycle. The governance process includes the ways to plan and execute the releases. This involves several activities, including:

- Aligning releases with strategic goals
- Validating releasable increments
- Ensuring compliance with standards and regulations
- Assessing customer impact
- Maintaining the supporting assets and activities for releasing

Release frequently and continually optimize the process ☰ Releasing frequently helps reduce time-to-market. Additionally, establishing successful continuous delivery and governance processes is only possible when the releases happen on a frequent, reliable basis. Over time, solution assets, architectures, and the infrastructure evolve and accumulate technical debt that may unexpectedly disrupt the release process. Releasing regularly helps uncover, mitigate, or even prevent those issues before they cause damage.

Getting Feedback

Getting fast feedback is the primary component of an ART’s high development velocity: speed comes from fast learning and adaptation rather than from “working harder.” Technology feedback results from integration and testing as well as running technical spikes. The feedback on the product value comes from the customer and business stakeholders. ARTs routinely:

Involve the customer in the development process ☰ There is no substitute for direct customer input. Including it in a routine development process helps an ART move at a much higher speed to avoid the costly mistake of building capabilities the customer doesn't need or cannot use. Preparation for PI planning, the PI planning itself, and system demos provide venues for customer interaction.

Measure business outcomes and usage ☰ Customer use of solutions may reveal issues and opportunities that otherwise might remain invisible to the ART. Creating the data capture and analytics capabilities, however, requires investment in the train's capacity, a proactive approach, and the use of [Architectural Runway](#). Additionally, an ART must measure whether delivered solutions enable the desired business outcomes—the ultimate purpose of the ART's effort.

Perform routine A/B testing ☰ Successful solution development is contingent upon an ART's ability to navigate the unknowns and make effective decisions. A/B testing enables effective decision-making and improves the development speed of an ART. Instead of prematurely committing to certain functionality, the ART creates two or more options and validates them with users, thus gaining a real sense of which alternative is performing better.

Test User Experience ☰ User Experience (UX) is essential to fully realizing the solution potential. But to provide productive UX, there needs to be an explicit, thorough UX design and testing strategy. As a part of this process, hypotheses are formulated, and then Minimum Marketable Features (MMF) are built and evaluated by observing the user in action, surveying them, or utilizing analytics. The SAFe [Lean UX](#) article covers additional topics of enabling effective UX.

Improving Relentlessly

An ART seeks to continuously improve productivity in delivering customer value. Naturally, the process requires measuring different aspects of ART operations and identifying areas for improvement:

Measure competency, flow, and outcomes ☰ Every ART should regularly assess against key applicable competencies. ARTs also should routinely measure [ART Flow](#) and apply [Flow Accelerators](#) to initiate forward momentum for ongoing flow improvement. Additionally, ARTs use their [Value Stream KPIs](#) to measure the outcomes that underpin the desired customer and business benefits.

Inspect & Adapt at regular intervals ☰ At every PI boundary, an ART has an opportunity to look back at the last PI, identify problems, and take corrective action during the [Inspect & Adapt](#) (I&A) event. This is the perfect time to identify significant, systemic improvement opportunities.

Make small improvements on the fly ☰ Every ART routinely discovers small, local, and tactical improvement opportunities. In most cases, it is best to address these as they occur and without waiting for the next I&A. This achieves quick wins and preserves the I&A for issues that require more attention and the involvement of high-

profile stakeholders.

Leverage Innovation & Planning Iteration ☰ The IP Iteration offers an opportunity to allocate uninterrupted time to innovation and learning. This helps the ART to further advance its solution, technical infrastructure, and various processes.

Learn More

[1] Skelton, Mathew, and Manuel Pais. *Team Topologies*. IT Revolution Press, 2019.

Last update: 24 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd, Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



While building trust gives teams the ability to reconfigure and Òdo the right thing,Ó it is also necessary to make sure that team members know what the right thing is. Team members must all work toward the same goal, and in volatile, complex environments that goal is changeable.

Adjust Cookie Settings

ÑGeneral Stanley
McChrystal, *Team of
Teams*

Team Backlog

The Team Backlog is a Kanban system that is used to capture and manage the user stories and enablers intended to enhance the solution.

This includes stories originating from Features in the ART backlog as well as those arising from the teamÕs local context.

Details

The team backlog holds all the possible work that a team might do to enhance the solution. For example, it contains User [Stories](#), [Enablers](#), and other work items such as improvement stories for corrective actions from the teamÕs [Retrospectives](#) or the [ARTÕs Inspect and Adapt](#).

While itÕs not conceptually complex, some essential aspects make the team backlog critical for Agile development. For example:

It contains all the work for an [Agile Team](#) to advance the solution and aligns all team members to a common goal.

ItÕs a list of wants, not commitments. Items can be estimated (preferable) or not, but it is just an ordered list, and there is no specific time commitment for completion. In other words, the backlog is time-independent, giving the team general flexibility regarding what gets implemented and when.

All team members can enter stories into the backlog.

It has an ownerÑthe Product Owner (PO)Ñwho helps the team manage the challenge of multiple stakeholders who may have divergent views of whatÕs essential.

The PO, with input from the team and other stakeholders, is primarily responsible for creating and maintaining the team backlog. However, any team member can enter an item into the backlog for consideration. The PO prioritizes the backlog, balancing the needs of stakeholders. There are three primary inputs to the team backlog, as Figure 1 illustrates.

Figure 1. Input sources for a team backlog

Figure 1. Input sources for a team backlog

ART Backlog ☰ The [ART Backlog](#) consists of upcoming features planned to be delivered by a train. During [PI Planning](#), teams split the candidate features into stories and tentatively place them into upcoming [Iterations](#). These new stories are maintained in the team backlog.

Team’s local context ☰ The team’s local concerns (other new functionality, defects, refactors, tech debt, and maintenance) are also in the backlog. Since PI planning is high-level, adjustments will likely occur during the PI. Teams using Scrum will probably make adjustments during [Iteration Planning](#), while teams applying Kanban will likely do the same during backlog replenishment.

Other stakeholders ☰ Agile Teams on the ART are not islands, and their backlogs will contain some stories that support other teams’ dependencies and other commitments, including the ART’s [PI Objectives](#). These stories may include spikes for research required to estimate [Features](#), [Capabilities](#), and even [Epics](#).

Moreover, teams get feedback from previous increments, the [System Demo](#), and other groups that may affect the backlog.

Nonfunctional Requirements (NFRs) are persistent qualities that may affect the solution's design, performance, or quality. Since they serve as constraints (or restrictions) for all the team's items, the Big Picture illustrates them at the bottom of the backlog in Figure 1. Due to their importance, teams often automate acceptance tests for NFRs and include them in their definition of done (DoD).

Building and Refining the Backlog

Agile Teams take a continuous, flow-based approach to maintain backlog readiness, so it always contains some stories ready for implementation without significant risk or surprise. Like a neglected garden that grows wild when left unattended for too long, the team backlog becomes unmanageable if not given care and attention. Refining the team backlog includes the following activities:

- Refining stories and establishing acceptance criteria
- The PO regularly prioritizes the team backlog in collaboration with the team and stakeholders
- New stories, including enablers, are discovered and described, and existing ones are changed or removed
- High-priority items are readied by defining acceptance criteria and sizing them to fit within small timeboxes
- Stories that have been around too long, or perhaps are no longer relevant, are removed

Although the PO manages the team backlog, refinement is a collaborative process. It creates a dialogue between the team, *Customers*, and other stakeholders. This refinement breaks down barriers between the business and the development team, eliminating waste, handoffs, and delays. Developing story acceptance criteria increases the clarity of the requirements, leverages the team's collective knowledge and creativity, and creates buy-in and joint ownership.

There is no prescriptive meeting pattern for refining the backlog. Some teams like to do a bit of backlog refinement after their Team Sync. Others prefer weekly refinement sessions or requirements specification workshops, applying *Behavior-Driven Development* (BDD) techniques to help clarify stories. Since multiple teams often collaborate on feature development, new issues, dependencies, and stories will likely arise. Backlog refinement also helps surface problems with the current plan, which may require discussion at the team, PO, or coach syncs.

Managing the Backlog with Kanban

In SAFe, Agile Teams manage their backlog using a Kanban system. The backlog

Kanban system facilitates alignment, visibility, and dependency management. Figure 2 illustrates an example of one team's initial Kanban system.

Figure 2. One Agile Team's initial Kanban board

Figure 2. One Agile Team's initial Kanban board

This Kanban visualizes all active and pending work, workflow states, and work-in-process (WIP) limits. The system is WIP limited; a work item can be pulled into the next step only when the number of items is lower than the WIP limit. A few activities in the Kanban (typically beginning and end) may not be WIP-limited. The team defines and adjusts WIP limits, allowing it to adapt quickly to the flow of complex system development variations.

See the [Applying Kanban in SAFe](#) and [SAFe Team Kanban](#) articles for more information on establishing the team Kanban system.

Balancing Value Delivery and System Health

with Capacity Allocation

Like the ART, every [Agile Team](#) faces the problem of balancing internal work—maintenance, refactors, and technical debt—with the new user stories that deliver more immediate business value. While focusing solely on business functionality may work for a while, this approach will be short-lived as technical debt increases, ultimately slowing development velocity. Avoiding this risk requires continuous investment in evolving the solution’s [Architectural Runway](#) while making customers happy with enhancements, new functionality, and bug fixes. Getting this balance right extends the system’s life, deferring technical obsolescence.

But prioritizing different types of work can be challenging as the PO tries to compare the value of unlike things: defects, refactors, redesigns, technology upgrades, and new user stories. And there is no upper limit to the demand for any of these things.

In collaboration with the team, the PO applies *capacity allocation* (Figure 3) for each item type. Then the PO, team, and [System Architect](#) select the highest-priority backlog items for each capacity allocation slice during planning. Since many stories originate from features, PI planning commitments may predetermine some priorities. However, the PO can prioritize work from the team’s local context by comparing value, size, and logical sequencing. Also, the PO can adjust the allocation percentage for each work item type to address long-term system health and value delivery. Teams should adapt the capacity allocation categories as needed. However, these categories should be consistent across teams in the ART.

Figure 3. Typical examples of capacity allocation categories (user stories, enablers, and maintenance in this case)

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*The people
who are crazy
enough to
think they can
change the
world are the
ones who do.*

Apple
1997 Think
Differently
Campaign

[Adjust Cookie Settings](#)

SAFe Practice Consultant

SAFe Practice Consultants (SPCs) are certified change agents who combine their technical knowledge of SAFe with an intrinsic motivation to improve the company's software, systems, and Agile business processes.

SPCs play a critical role in successfully implementing SAFe. They come from numerous internal or external roles, including business and technology leaders, portfolio/program/project managers, process leads, architects, analysts, and consultants.

Find a Course:

Go

Details

Business Agility is the ability to compete and thrive in the digital age by quickly responding to market changes and emerging opportunities with innovative, digitally-enabled business solutions. For traditional enterprises, recognizing the fact that success is now enabled by the ability to create these solutions is a bit of a "wake-up call" or even an existential crisis. Even for companies that were born in the digital age, future success cannot be assured. SAFe Practice Consultants (SPCs) work within and across organizations to provide new methods and practices that connect the entire enterprise with the principles that enable Business Agility. This critical role answers a vital need to build the future of work.

To achieve meaningful and lasting change, author John P. Kotter notes that stakeholders need a "sufficiently powerful guiding coalition". [1] Such a coalition requires:

- Leaders who can set the vision, show the way, and remove impediments
- Practitioners, managers, and change agents who can implement specific process changes
- Sufficient organizational credibility to affect change
- The expertise needed to make fast, intelligent decisions

In enterprises utilizing SAFe, these coalitions require experienced and trained SPCs.

Responsibilities

As knowledgeable change agents, SPCs have a primary role in many of the 13 critical moves described in the [SAFe Implementation Roadmap](#). They also must work beyond the roadmap as experts and coaches that help the organization achieve business agility. Doing so requires fulfilling a broad set of responsibilities, as seen in Figure 1.

Figure 1. SAFe Practice Consultant Responsibilities

Figure 1. SAFe Practice Consultant Responsibilities

Embodying a Lean-Agile Mindset

SPCs lead change conversations across multiple business departments and organizational hierarchies. To be credible and effective, they must speak with knowledge and competence. In addition, to inspire changes in the behaviors of others, SPCs first create change in themselves. Required mastery and knowledge includes:

Exhibit the Lean-Agile Mindset ☰ To help others achieve a [Lean-Agile Mindset](#), each SPC must lead the way. This will be a continuous journey each SPC must choose to take. SAFe SPCs model the values and principles of Lean Thinking and the Agile Manifesto in their daily interactions.

Demonstrate the SAFe Core Values ☰ By exemplifying respect for people, alignment, transparency, and relentless improvement, SPCs demonstrate a common value system and inspire others to the same beliefs. SPCs feel a personal accountability to live the core values and lead by example.

Utilize the SAFe Principles ☰ SPCs have internalized the [SAFe Principles](#) and know how to use them to initiate and sustain conversations that inspire change. They sense the issues hindering agility and apply their understanding to clearly communicate the principles that inform transformed behavior and practices.

Leading the Change

SPCs know the “job” that must be done to achieve the change vision. They communicate the business need, urgency, and vision for change with the assistance of critical SAFe training and assets that are purpose-built to provide a proven path to organizational change.

SPCs know how to apply the right asset at the right moment, utilizing the full breadth of SAFe as their implementation toolkit. Change activities include:

Communicate the vision for change ☰ SPCs know that change involves subtle overlapping stages. They guide teams at multiple levels throughout the organization, handling the cycles of change and communicating the vision as each emerges. To communicate clearly and effectively, SPCs partner with enterprise leaders to articulate the path forward and to coach leaders. When change is not flowing through the organization, however, SPCs identify opportunities for improvement.

Maintain the Guiding Coalition ☰ As the transformation grows, so does the need for the guiding coalition. SPCs assist the organization in building and growing the LACE and Value Management Office (VMO) and help identify the proper individuals for those organizations. SPCs also encourage emerging members of both groups to become SPCs to understand the transformation fully. New leaders will emerge throughout all layers as the organization becomes more transparent. SPCs identify and align them into a growing network of connected guiding coalition teams. (see the Extended Guidance [article](#) for an example)

Maintain the transformation backlog ☰ SPCs throughout the organization, the LACE, and the VMO, align on transformation backlogs, enabling a shared vision for the changes required to improve flow, advance skills, grow role-based competencies, grow mindset, and more. Teams, trains, LACEs, and Portfolios all have local improvements. SPCs enable alignment through active communication and energy that mitigates complacency and silo-based optimizations. SPCs encourage experimentation with practices to gain the right wins for the opportunities at hand, applying SAFe’s Principles, Values, and Lean Thinking to keep experimentation

connected.

Launch Exemplar COPs As the SPC community grows through supporting an organization, they will work together to identify their strengths. SPCs often form a CoP to share knowledge and support each other. SPCs also help create and enhance other CoPs, enabling collaborative environments that furthers the organizational ability to problem-solve, learn and retain top talent.

Implementing SAFe

The implementation roadmap assists SPCs in the tactical steps required to lead the change, ensuring that each Ônext stepÕ is done without ignoring previous changes that may still require attention. Treating every implementation as if it were their first, SPCs transfer their enthusiasm, energizing the new participants.

In so doing, SPCs apply and often lead many of the critical moves of [the SAFe Implementation Roadmap](#) as described below:

[Reaching the tipping point](#) Ø They communicate the business need, urgency, and vision for change.

[Create a Lean-Agile Center of Excellence \(LACE\)](#) Ø SPCs assist the LACE with building and executing the transformation backlog.

[Training executives, managers, and leaders](#) Ø They socialize the new concepts and provide orientation and overview training, specifically teaching all directors, managers, executives and leaders associated with the transformation in Leading SAFe.

[Lead in the Digital Age](#) Ø SPCs facilitate modules of the Leading in the Digital Age series for leaders, managers, and stakeholders, continuing after these modules to help leaders form and practice new behaviors and methods.

[Organize around value](#) Ø Working with stakeholders to understand the flow of value, SPCs facilitate the identification of [Operational Value Streams](#), [Development Value Streams](#), [ARTs](#) and [Portfolios](#) to find those that are the most opportunistic for launch.
[Create the implementation plan](#) Ø SPCs participate in creating a plan for the rollout, communicate upcoming changes, and establish metrics.

[Prepare for the ART launch](#) Ø SPCs help the LACE plan and prepare for the ART launch. They coach leadership and help facilitate the creation of new [Agile Teams](#). They also train or source training of executives, leaders, Agile teams, and specialty rolesÑsuch as Product Owner, Product Manager, Scrum Master, and Release Train Engineer (RTE). They also assess and evolve launch and [backlog](#) readiness.

[Train teams and launch the ART](#) Ø SPCs often directly plan and execute ÔquickstartÕ or other rollout strategies. They train or source training for teams and participate in initial, critical events like [PI Planning](#) and [Inspect and Adapt \(I&A\)](#). SPCs help establish the ART launch date and calendar for ART and Team events.

Coach ART execution Ⓛ The SPCs coach leaders and stakeholders to build and maintain the [Vision](#), [Roadmap](#) and [backlogs](#). They coach teams, Product Owners, Product Managers, Architects, and RTEs. They guide the shift from project-to-product with a focus on Customer Centricity and Design Thinking as part of Agile Product Delivery. They also participate in Scrum of Scrums and System Demo, facilitate I&A and follow-up on improvement items. Finally, they help teams establish a DevOps culture and mindset, the [Continuous Delivery Pipeline](#), infrastructure, and associated [built-in quality](#) practices.

Launch more ARTs and value streams Ⓛ SPCs work to enable new change agents to increase organizational capacity to support new value streams, start more ARTs, and expand the reach of the LACE. They communicate progress and highlight early accomplishments.

Enhance the portfolio Ⓛ Once Lean-Agile practices gain momentum, SPCs can socialize and optimize those practices in the portfolio level, including Portfolio Vision, Lean Budgets using Participatory Budgeting and [Guardrails](#) and [Lean Portfolio Management](#).

Accelerate Ⓛ An enterprise's SAFe journey doesn't end with the launching of trains and the adoption of Lean Portfolio Management. SPCs have expert mastery of the seven competencies of the Lean Enterprise and how they contribute to achieving Business Agility.

Coaching Flow

SPCs have a critical role in enabling all members of an organization to understand the flow of value and how to [Make value flow without interruption \(SAFe Principle 6\)](#).

Coaching flow effectively requires SPCs throughout the organization to:

Facilitate value stream mapping Ⓛ SPCs facilitate [Value Stream Mapping](#) exercises to identify and maintain the overall

flow efficiency of a full value stream or portion of a value stream. They use data from the teams, the tools, and the outcomes of improvement items to maintain the value stream maps, improving the continuous delivery pipeline.

Establish the Kanban systems Ⓛ SPCs help teams understand and utilize [Kanban](#) boards and practice through each layer of SAFe, including Agile Teams, ARTs, Solution Trains, and Portfolios. SPCs help create connections throughout the Kanban systems, enabling strategy to flow through execution.

Measure flow Ⓛ SPCs understand [flow metrics](#) and how they work alongside competency and outcome metrics to facilitate enterprise flow improvements. They help the organization to use quantitative and qualitative data to validate flow improvements and inform further steps. SPCs help create routine leadership participation in improvement activities as well as a certain degree of decentralized decision-making to enable the knowledge workers to directly improve their work

environment and achieve faster flow, encouraging transparent-to-all measurement along the way.

Apply the Flow Accelerators Ⓜ The SPC utilizes Flow Accelerators to improve flow performance in the context of their organization. Applying the Flow Accelerators has significant differences depending on the level at which they are applied: [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#). The SPC has many opportunities available across various SAFe events, workshops and trainings to accelerate flow ongoing.

Grow a flow mindset Ⓜ SPCs actively participate in community discussions on improving flow through the system. They bring facts to conversations, enabling flow visualization and resolving impediments. Making flow visible is a multi-faceted effort that involves extensive use of flow metrics, information radiators, tooling, and productive communication around the issues of flow. SPCs foster the interactions between leaders and teams, establishing healthy visibility and Gemb. SPCs use 1:1 and small group discussions consistently to grow the flow mindset outside of workshops and events as the organization evidences the wins that the flow improvements introduce to their value delivery. Gradually, a shared mental model of flow is established that helps the organization navigate through their journey of flow optimization and transformation.

Accelerating Business Agility

SPCs keep the long-term business agility vision moving. They enable business agility, not by over-optimizing a piece of the system, but by optimizing the system as a whole. Additionally, SPCs measure and cultivate competencies at all levels, run workshops, foster training opportunities, and assist new SPCs in their roles. SPCs sense and create opportunities to:

Extend agility across the business Ⓜ Organizations may begin their SAFe journey within a specific business department, usually within IT, Cyber-physical, or Digital departments. SPCs find opportunities to apply patterns of agility throughout their organization, fostering new areas of transformation. Utilizing [Business and Technology patterns](#) of those who have come before and identifying and sharing new patterns, they help their entire enterprise in achieving full business agility.

Measure and Grow the core competencies Ⓜ SPCs help companies achieve business agility by [measuring and growing](#) the core competencies at all levels. They work with other SPCs to align up and down and across the enterprise to create systemic improvement tidal waves. They coach competency improvements, maturing others to be able to simplify the inherent complexity that comes with scale.

Expand the SPC pool Ⓜ As the SPC pool grows, seasoned SPCs assist new SPCs in building their confidence and understanding while supporting the transformation. To achieve the breadth and depth of Agile leadership throughout the organization as

the transformation grows, SPCs must enable other SPCs.

Sustain the change ☰ As needed, SPCs will continue to lead *Value Stream and ART Identification* workshops and discussions to organize and reorganize around value.

Partnering with Lean-Agile leaders throughout the organization, they'll continue to advance the guiding coalition and the change process. They also participate in and lead trainings and workshops to enable business growth, including design thinking workshops, epic writing, story mapping, and more. While maintaining the foundation that started during the first Agile experiments, they help apply test-and-learn techniques to new spaces.

The tactical items above will comprise the new way of working, with cadence and alignment becoming habits. SPCs will continue to keep new ways of working meaningful and flow based. SPCs know how to listen for the signals in their organization that indicate stunted growth and improvement. They use these signals to pull people together across solutions, creating transparency toward shifting conversations from simply tracking work to strategic decision-making. SPCs must also be aware that this is a value and principle based system, and that events and activities must stay activated. The new thinking must be anchored by SPCs and the larger guiding coalition in the mindset and culture of the organization. Without this, teams and leaders will invariably fall back into old behaviors. Multiple roles in the organization maybe SPCs, and each are critical to this continuous involvement.

How many SPCs do you need

At first glance, the responsibilities above seem daunting. No single SPC could accomplish all this alone, as the knowledge and skills of an SPC can't be limited to a few select people within a scaled enterprise. For example, an SPC who coaches executives often differs from an SPC who coaches technical practices to Agile teams. Instead, many leaders across the emerging Lean-Agile business must master these distinctive new competencies. This means most companies will require many SPCs (as many as 5+ per 100 practitioners) to drive and sustain the implementation.

Achieving SPC training and certification

SPCs must be trained for their new role and certified to be enabled to train others. They need to acquire the skills and tools to execute their responsibilities and coach and teach others to implement and support the change. The best way to achieve this is to take the [Implementing SAFe with SAFe® Practice Consultant \(SPC\) certification](#) class. This four-day course prepares SPCs

to become the change agents who lead the transformation. Attendees learn how to effectively apply the principles and practices of SAFe and organize, train, and coach Agile teams. They also learn to identify value streams, identify and launch ARTs, and help build and manage an Agile portfolio. To make the role-based training required across the organization practical and cost-effective, [Scaled Agile, Inc.](#) supports a train-the-trainer, fan-out model, licensing SPCs to teach SAFe courses that support the other key roles in the implementation. This provides an affordable training strategy and supplies the trainers needed to achieve the mission of company-wide change.

I'm an SPC. Now What?

After passing an exam, attendees become certified SPCs, gaining access to various helpful SPC resources to facilitate SAFe adoption. They will also be licensed to teach the courses listed [here](#).

Learn More

[1] Kotter, John P. *Leading Change*. Harvard Business Review Press, 1996.

Last update: 6 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Many leaders pride themselves on setting the high-level direction and staying out of the details. But big picture, hands off leadership isn't likely to work in a change situation, because the hardest part of change—the paralyzing part—is in the details.

Any successful change requires a translation of ambiguous goals into concrete behaviors. To make a switch, you need to script the critical moves.

—Dan and Chip Heath, *Switch: How to Change Things When Change Is Hard*

[Adjust Cookie Settings](#)

SAFe Implementation Roadmap

Note: This is the **home page** for the **SAFe™ Implementation Roadmap** series. There are an additional 13 articles in this series, each of which is linked in sequence from the sections below.

The SAFe Implementation Roadmap consists of an overview graphic and a 14-article series that describes a strategy and an ordered set of activities for successfully implementing SAFe.

Achieving business agility and the benefits of Lean-Agile development at scale is not a trivial effort, so SAFe is not a trivial Framework. Before realizing SAFe's rewards, organizations must embrace a Lean-Agile Mindset and understand and apply Lean-Agile principles. They must identify their Portfolio(s), Value Streams, and Agile Release Trains (ARTs), implement their Lean-Agile Portfolio(s), build quality in, and establish the mechanisms for continuous flow of value delivery and DevOps. And, of course, the culture must evolve as well.

The transformation of an organization towards full business agility requires the change leadership acumen to create persistent people practices and the mindset needed to support the change. Such a change also requires a practical and proven guide that brings the change into sequential action supported by a robust guiding coalition. Based on proven organizational change management strategies, the SAFe Implementation Roadmap graphic and article series describes the critical moves an enterprise can take to implement SAFe in an orderly, reliable, and successful fashion. As described in the Implementation Roadmap, the strategy for Implementing SAFe is based partly on John Kotter's work in organizational change management [1]. As listed below, Kotter identifies eight "Accelerators" of leading a change. These are integrated throughout the SAFe implementation roadmap to enable change leaders along their journey.

- Create a sense of urgency
- Build a guiding coalition
- Form a strategic vision
- Enlist a volunteer army
- Enable action by removing barriers
- Generate short-term wins
- Sustain acceleration
- Institute Change

Getting Started with Implementing SAFe:

Details

To achieve the desired organizational change, leadership must "script the critical moves," as described by Dan and Chip Heath [2]. When identifying those critical moves for adopting SAFe, hundreds of the world's largest enterprises have already gone down this path, and successful adoption patterns have emerged. Figure 2 illustrates a typical pattern.

Figure 2. SAFe Implementation Roadmap

While no two adoptions are identical, and there is rarely a perfectly sequential step-by-step implementation, we know that businesses getting the best results follow a path similar to that shown in the Implementation Roadmap. Each step on the roadmap is described in a separate article:

1. [Reaching the Tipping Point](#)
2. [Train Lean-Agile Change Agents](#)
3. [Create a Lean-Agile Center of Excellence](#)
4. [Train Executives, Managers, and Leaders](#)
5. [Lead in the Digital Age](#)
6. [Organize Around Value](#)

7. [Create the Implementation Plan](#)
8. [Prepare for ART Launch](#)
9. [Train Teams and Launch ART](#)
10. [Coach ART Execution](#)
11. [Launch More ARTs and Value Streams](#)
12. [Enhance the Portfolio](#)
13. [Accelerate](#)

This first article is a launching pad to explore the roadmap and understand how to apply it to specific implementations. The remaining articles will help you throughout the journey.

Let's get started with "Implementing" with the first article: [Reaching the Tipping Point](#).

NEXT

Learn More

[1] Kotter, John P. *Accelerate*. Harvard Business Review Press, 2014

[2] Heath, Chip, and Heath, Dan *Switch: How to Change Things When Change Is Hard*. Crown Publishing Group, 2010

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit [Permissions FAQs](#) and contact us for permissions.

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California Consumer Rights](#)

English

Scaled Agile Framework



*The impression that
our problems are
different is a
common disease
that afflicts
management the
world over. They are
different, to be sure,
but the principles
that will help to
improve the quality
of product and
service are universal
in nature.*

Adjust Cookie Settings

~W. Edwards

SAFe Lean-Agile Principles

SAFe is based on ten immutable, underlying Lean-Agile principles. These tenets and economic concepts inspire and inform the roles and practices of SAFe.

Figure 1. SAFe Lean-Agile Principles

Why the Focus on Principles?

Building enterprise-class software and cyber-physical systems are among the most complex challenges our industry faces today. And, of course, the enterprises that build these systems are also increasingly sophisticated. They are bigger and more distributed than ever. Mergers and acquisitions, distributed multinational (and multilingual) development, offshoring, and rapid growth are all part of the solution.

But they're also part of the problem.

Fortunately, we have an amazing and growing body of knowledge that can help. It includes Agile principles and methods, Lean and systems thinking, product development flow practices, and Lean processes. Thought leaders have traveled this path before us and left a trail in hundreds of books and references to draw on.

The goal of SAFe is to synthesize this body of knowledge, along with the lessons learned from hundreds of deployments. This creates a system of integrated, proven practices that have improved employee engagement, time-to-market, solution quality, and team productivity. Given the complexities, however, there's no off-the-shelf solution for the unique challenges each enterprise faces. Not every SAFe recommended practice will apply equally in every circumstance. This is why we work hard to ensure that SAFe practices are grounded in fundamentally stable principles. That way, we can be confident the practices apply in most situations.

And if those practices do fall short, the underlying principles will guide the teams to make sure that they are moving continuously on the path to the goal of the lean: Øshortest sustainable lead time, with best quality and value to people and society.Ø There is value in that, too.

SAFe is based on ten fundamental concepts that have evolved from Agile principles and methods, Lean product development, systems thinking, and observation of successful enterprises. Each is described in detail in an article by that principle's name. In addition, the embodiment of the principles appears throughout the Framework. They are summarized in the following sections, and each has a full article behind the link.

#1 Ø Take an economic view

Delivering the Øbest value and quality for people and society in the shortest sustainable lead timeØ requires a fundamental understanding of the economics of building systems. Everyday decisions must be made in a proper economic context. This includes the strategy for incremental value delivery and the broader economic framework for each value stream. This framework highlights the trade-offs between risk, Cost of Delay (CoD), manufacturing, operational, and development costs. In addition, every development value stream must operate within the context of an approved budget and be compliant with the guardrails which support decentralized decision-making.

#2 Ø Apply systems thinking

Deming observed that addressing the challenges in the workplace and the marketplace requires an understanding of the systems within which workers and users operate. Such systems are complex, and they consist of many interrelated components. But optimizing a component does not optimize the system. To improve, everyone must understand the larger aim of the system. In SAFe, systems thinking is applied to the system under development, as well as to the organization that builds the system.

#3 ⚡ Assume variability; preserve options

Traditional design and life cycle practices encourage choosing a single design-and-requirements option early in the development process. Unfortunately, if that starting point proves to be the wrong choice, then future adjustments take too long and can lead to a suboptimal design. A better approach is to maintain multiple requirements and design options for a longer period in the development cycle. Empirical data is then used to narrow the focus, resulting in a design that creates optimum economic outcomes.

#4 ⚡ Build incrementally with fast, integrated learning cycles

Developing solutions incrementally in a series of short iterations allows for faster customer feedback and mitigates risk. Subsequent increments build on the previous ones. Since the Ÿsystem always runs,Ó some increments may serve as prototypes for market testing and validation; others become minimum viable products (MVPs). Still others extend the system with new and valuable functionality. In addition, these early, fast feedback points help determine when to ÔpivotÓ where necessary to an alternate course of action.

#5 ⚡ Base milestones on objective evaluation of working systems

Business owners, developers, and customers have a shared responsibility to ensure that investment in new solutions will deliver economic benefits. The sequential, phase-gate development model was designed to meet this challenge, but experience shows that it does not mitigate risk as intended. In Lean-Agile development, integration points provide objective milestones at which to evaluate the solution throughout the development life cycle. This regular evaluation

provides the financial, technical, and fitness-for-purpose governance needed to ensure that a continuing investment will produce a commensurate return.

#6 Ⓛ Make value flow without interruptions

The fourth principle in Lean Thinking is to Ômake value flow without interruptions.Õ Doing so requires an understanding of what flow is, what the various properties of a flow system are, and how these properties can accelerate or impede the flow of value through any particular system. Principle #6 highlights the eight common properties of a flow-based system and provides specific recommendations for eliminating impediments to flow.

#7 Ⓛ Apply cadence, synchronize with cross-domain planning

Cadence creates predictability and provides a rhythm for development. Synchronization causes multiple perspectives to be understood, resolved and integrated at the same time. Applying development cadence and synchronization, coupled with periodic cross-domain planning, provides the mechanisms needed to operate effectively in the presence of inherent development uncertainty.

#8 Ⓛ Unlock the intrinsic motivation of knowledge workers

Lean-Agile leaders understand that ideation, innovation, and employee engagement are not generally motivated by individual incentive compensation. Such individual incentives can create internal competition and destroy the cooperation necessary to achieve the larger aim of the system. Providing autonomy and purpose, minimizing constraints, creating an environment of mutual influence, and better understanding the role of compensation are keys to higher levels of employee engagement. This approach yields better outcomes for individuals, customers, and the enterprise.

#9 Ⓛ Decentralize decision-making

Achieving fast value delivery requires decentralized decision-making. This reduces delays, improves product development flow, enables faster feedback, and creates

more innovative solutions designed by those closest to the local knowledge. However, some decisions are strategic and global and have economies of scale that justify centralized decision-making. Since both types of decisions occur, creating a reliable decision-making framework is a critical step in empowering employees and ensuring a fast flow of value.

#10 ⚡ Organize around value

Many enterprises today are organized around principles developed during the last century. In the name of intended efficiency, most are organized around functional expertise. But in the digital age, the only sustainable competitive advantage is the speed with which an organization can respond to the needs of its customers with new and innovative solutions. These solutions require cooperation amongst all the functional areas, with their incumbent dependencies, handoffs, waste, and delays. Instead, Business Agility demands that enterprises organize around value to deliver more quickly. And when market and customer demands change, the enterprise must quickly and seamlessly reorganize around that new value flow.

Last update: 11 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Linkedin](#)

[YouTube](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Agility is principally about mindset, not practices.

— Jim
Highsmith, Agile
Project Management:
Creating Innovative
Products

Lean-Agile Mindset

The Lean-Agile Mindset is the combination of beliefs, assumptions, attitudes, and actions of SAFe leaders and practitioners who embrace the concepts of Lean

Thinking and the Agile Manifesto.

It's the personal, intellectual, and leadership foundation for adopting and applying SAFe principles and practices.

The Lean-Agile mindset forms the cornerstone of a new way of working and an enhanced company culture that enables Business Agility. It provides leaders and change agents with the tools needed to drive a successful SAFe transformation, helping individuals and enterprises achieve their goals.

Details

Mindset Awareness and Openness to Change

What exactly is a "mindset"? A mindset is a mental lens through which we view the world around us. It is how the human brain simplifies, categorizes, and interprets the vast amount of information it receives daily. We form our mindsets through a lifetime of structured learning (classes, reading) and unstructured lessons (life events, work experience). They reside in the subconscious mind and manifest themselves as deeply held beliefs, attitudes, assumptions, and influences. Consequently, individuals are often unaware of how their mindsets influence how they carry out their responsibilities and interact with others. While many mindsets are positive and serve us well, others may need to change over time [1].

So how can mindsets be changed? It begins with an awareness of one's current mindsets and how they were formed. It's also vital to cultivate the belief that mindsets can be developed and improved (a "growth" mindset, as illustrated in Figure 1).

Figure 1. Adopting a new mindset requires a belief that new abilities can be developed with effort

Changing mindsets is a vital topic in transitioning to SAFe because, too often, leaders and practitioners in organizations go through the motions of mimicking SAFe practices and using SAFe terms without internalizing and embracing the underlying values and principles that truly represent a new way of working. This ÔSAFe in name onlyÕ approach may produce some small successes in the short term. However, in the long term, such a shallow adoption of the Lean-Agile mindset will inevitably fail to produce the real, long-lasting business results leaders hoped for when they decided to Ôgo SAFe.Õ

To fully embrace SAFe requires a ***growth*** mindset open to learning the core values and principles of two primary underlying bodies of knowledge: Lean Thinking and Agile. Each has a rich and deep history of published guidance and case studies. Their respective values and principles need to be understood and practiced so that the ideals of both Lean and Agile permeate the organizationÕs language, practices, and decision-making. Ultimately, it simply becomes Ôour way of workingÕ and is deeply ingrained in the culture of the enterprise.

The following two sections describe the key elements of Lean Thinking and Agile (summarized in Figure 2) that form the basis of the Lean-Agile mindset.

Figure 2. Lean Thinking and Agile are the core building blocks of SAFe

Figure 2. Lean Thinking and Agile are the core building blocks of SAFe

Lean Thinking

// Adopting Lean product development can double labor productivity through the entire system, cut time-to-market for new products in half, and enable a wider variety of products within product families to be offered at very modest additional cost.

Ñ from *Lean Thinking* by Womack & Jones

Initially derived from Lean manufacturing, the principles and practices of Lean thinking as applied to software, product, and systems development are now deep and extensive [2]. For example, Ward [3], Reinertsen [4], Poppendieck [5], Kersten [6], Leffingwell [7], and others have described aspects of Lean thinking, placing many of the core principles and practices within a product development context. Applying Lean Thinking to product development, thereby shifting from the traditional batch-and-queue production system to continuous flow with an effective pull by the customer, can lead to dramatic improvements.

Lean thinking can be summarized as shown in Figure 3:

Figure 3. The core tenets of Lean Thinking

Figure 3. The core principles of Lean Thinking

As Figure 3 illustrates, the goal of Lean Thinking is to deliver the maximum value (a solution) to the customer in the shortest sustainable lead time from the trigger (the identification of the need or opportunity) to the point at which the customer receives the value. How value is created also matters. High quality, respect for people and society, high morale, safety, and customer delight are also essential goals and benefits of Lean Thinking. Achieving these goals requires applying the five basic principles of Lean, illustrated in Figure 3 and described in the following sections.

Precisely specify value by specific product

Every enterprise is built to deliver value. Value can only be defined by the ultimate customer. And it's only meaningful when expressed in terms of a specific product (a good or a service, and often both at once) that meets the customer's needs at a specific price at a specific time [8]. Therefore, the first principle of Lean Thinking underscores the importance of understanding customers' needs and quantifying the value inherent in the solutions delivered to them. The solution itself holds the value—not the project, initiative, or process that produces it—and the customer ultimately determines that value.

Identify the Value Stream for each product

Once "value" is defined for each product and type of customer, the following principle in Lean Thinking is to articulate how the enterprise creates that value, from identifying a need or opportunity to delivering the solution. This flow of work is the value stream and contains all the people, processes, tools, and information necessary to deliver value. Delays anywhere in this system result in delayed delivery of value to customers.

Make value flow without interruptions

The third principle in Lean Thinking is establishing a continuous, uninterrupted flow of work that supports incremental value delivery based on constant feedback and adjustment. Enabled by [Built-In Quality](#) practices, relentless improvement, and evidence-based governance, continuous flow enables faster, sustainable value delivery.

Achieving a continuous flow of value requires applying and understanding the eight fundamental properties of flow: visualizing and limiting work-in-process (WIP), addressing bottlenecks, minimizing handoffs and dependencies, getting fast feedback, working in small batches, managing queue lengths, optimizing time "in the zone," and remediating legacy policies and practices. These flow properties are described in greater detail in the [SAFe Principle 6](#) article and in the [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#) articles. The SAFe [Core Values](#) and [SAFe Principles](#) help teams achieve a continuous flow of value at scale in large, complex enterprises.

Let the customer pull value from the producer

The next Lean principle guides organizations to configure value streams to deliver solutions that customers *pull* into the market based on their actual needs rather than solutions that teams *push* into the market based on what they "think" customers need. This is key to calibrating the capacity of the value stream. Too much capacity compared to market pull results in waste, which is antithetical to

Lean thinking. Conversely, too little capacity creates bottlenecks and delays, defeating the intent to provide the customer with a continuous flow of value.

Pursue perfection

The final principle of Lean Thinking is expressed as “pursue perfection.” It reflects that no matter how closely the first five principles are followed, creating a fast and effective flow of value is not a one-time activity. Market dynamics, customer needs, and available technologies are just some of the many factors that can require value streams to be refined and, in some cases, rebuilt entirely.

Agile



Agile is an attitude, not a technique with boundaries. An attitude has no boundaries, so we wouldn't ask “can I use Agile here,” but rather “how would I act in the Agile way here,” or “how Agile can we be, here?”

— Alistair Cockburn

In the 1990s, some lighter-weight and more iterative development methods emerged in response to the many challenges of waterfall processes. In 2001, many leaders of these frameworks came together to express their shared values and beliefs in the Manifesto for Agile Software Development. This turning point clarified the new approach and started to bring the benefits of these innovative methods to the whole development industry. [8] Since the Manifesto was first published, Agile has been adopted by domains outside of software development, including hardware systems, infrastructure, operations, and support. More recently, business teams outside of technology have also embraced Agile principles for planning and executing their work.

The Values of Agile

Agile is built on the value statement shown in Figure 4. For the remainder of the description of Agile Values and Principles, as initially outlined in the Agile Manifesto, readers can expand each use of the term “software” to include the working output of any Agile team, regardless of the domain.

Figure 4. Agile Values

Figure 4. Agile Values

Individuals and Interactions over Processes and Tools

Deming notes, “If you can’t describe what you are doing as a process, then you don’t know what you are doing.” [10] So, Agile processes in frameworks like Scrum, Kanban, and SAFe matter. However, a process is only a means to an end. When we’re captive to a process that isn’t working, it creates waste and delays. So, favor individuals and interactions, then modify processes accordingly. Tools are valuable but should supplement, rather than replace, face-to-face communication.

Working Software over Comprehensive Documentation

Documentation is important and has value. But creating documents to comply with potentially outdated corporate governance models has limited value. As part of a change program, governance, often captured by documentation standards, needs to be updated to reflect the Lean-Agile way of working. Rather than create detailed documentation too early—especially the wrong kind—it’s more valuable to show customers working software, systems, and so on to get their feedback. Therefore, favor measuring progress by evaluating tangible work products. And document only what’s truly needed.

Customer Collaboration over Contract Negotiation

Customers are the ultimate deciders of value, so their close collaboration is

essential in pursuing business agility. Contracts are often necessary to convey each party's rights, responsibilities, and economic concerns—but recognize that contracts can over-regulate what to do and how to do it. They don't replace regular communication, collaboration, and trust, no matter how well they're written. Instead, contracts should be win-win propositions. Win-lose contracts usually result in poor economic outcomes and distrust, creating contentious short-term relationships instead of long-term business partnerships. Instead, favor customer collaboration over contract negotiation.

Responding to Change over Following a Plan

Change is a reality in the digital age and essential to achieving agility. The strength of Lean-Agile is in how it embraces change. As the system evolves, so does the understanding of the problem and the solution domain. Business stakeholder knowledge also improves over time, and customer needs evolve as well. Indeed, those changes add value to our system.

Of course, planning is an essential part of Agile. In fact, Agile teams and teams-of-teams plan more often and more continuously than their waterfall counterparts. However, plans must adapt as new learning occurs, new information becomes visible, and the situation changes. Worse, evaluating success by measuring conformance to a plan drives the wrong behaviors (for example, following a plan in the face of evidence that the plan is not working).

Agile Principles

Agile has 12 self-explanatory principles that support its values, as shown in Figure 5. These principles take Agile Values a step further and specifically describe what it means to be Agile.



Figure 5. Principles of the Agile Manifesto

The combination of these values and principles creates the essence of Agile. There is overwhelming evidence from success stories in all industries across every geography demonstrating the extraordinary business and personal benefits of this new way of thinking and working. We are grateful for it.

Applying Lean Thinking and Agile in SAFe

Collectively, the values and principles of Lean Thinking and Agile form the DNA of everything contained within SAFe. All the roles, practices, events, and artifacts in SAFe are designed to provide practical guidance for adopting the combination of these two bodies of knowledge as the new way of working throughout the enterprise.

Thousands of implementations of SAFe over the last decade have shown that Lean Thinking and Agile principles and practices have unique implications when applied at scale. For example, providing an uninterrupted flow of value within the context of a single Agile team will look different than when this same principle is applied to an entire portfolio. Yet the principle is equally important in both cases. The

implications of Lean and Agile at scale have been captured in the SAFe Core Values and SAFe Principles articles.

Learn More

- [1] Dweck, Carol S. *Mindset: The New Psychology of Success*. Random House Publishing, 2007.
- [2] Womack, James P., Daniel T. Jones, and Daniel Roos. *The Machine That Changed the World: The Story of Lean Production—Toyota’s Secret Weapon in the Global Car Wars That Is Revolutionizing World Industry*. Free Press, 2007.
- [3] Ward, Allen, and Durward Sobeck. *Lean Product and Process Development*. Lean Enterprise Institute, 2014.
- [4] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas, 2009.
- [5] Poppendieck, Mary, and Tom Poppendieck. *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley, 2006.
- [6] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. IT Revolution Press, 2018.
- [7] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.
- [8] Womack, James, and Daniel Jones. *Lean Thinking: Banish Waste and Create Wealth in your Corporation*. Free Press, 2003.
- [9] Manifesto for Agile Software Development. <http://AgileManifesto.org/>
- [10] Deming, W. Edwards. *Out of the Crisis*. MIT Center for Advanced Educational Services, 1982.

Last update: 6 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Find people who
share your values,
and you'll conquer
the world together.*

— John Ratzenberger

Core Values

The four Core Values of alignment, transparency, respect for people, and relentless improvement represent the foundational beliefs that are key to SAFe's effectiveness.

Adjust Cookie Settings

These tenets help guide the behaviors and actions of everyone participating in a SAFe portfolio. Those in positions of authority can help the rest of the organization embrace these ideals by exemplifying these values in their words and actions.

Details

SAFe is a system for achieving business agility based on highly respected bodies of knowledge: Agile, Lean, systems thinking, DevOps, and value stream management, to name a few. Their relevance to business agility has been proven through the successful adoption of these practices by the world's largest organizations. These comprehensive success patterns make SAFe broad, deep, and scalable. But at its core, SAFe places the highest value on four deeply held beliefs: *alignment, transparency, respect for people, and relentless improvement*. These tenets are so fundamental to the practice of SAFe that without them, the practices in the Framework will inevitably fail to deliver the intended business results that prompted the decision to Ògo SAFe.Ó These core values are illustrated in Figure 1 and described in the following sections.

Figure 1. SAFe's four core values

Alignment

Like cars out of alignment, misaligned companies can develop serious problems. They are hard to steer and don't respond well to changes in direction [1]. Even if it's clear where everyone thinks they're headed, the vehicle is unlikely to get them there.

The same is true for organizations adopting SAFe as their new way of working. In Lean-Agile, many decisions are decentralized to deliver value in the shortest sustainable lead time (see SAFe Principle #9). However, if decisions pull the organization in different directions, significant delays and quality concerns will result. The solution is to provide clear, consistent alignment from the top of the enterprise through every level of SAFe, all the way to each individual contributor. Value delivery with speed and quality can consistently be achieved when everyone is aligned.

Here are specific ways to create and maintain alignment in SAFe.

Communicate the vision, mission, and strategy. Alignment starts with keeping the enterprise's vision, mission, and strategy constantly present. For example, including these elements in the Business Owner briefings during PI planning is one way to ensure the work of the ART is consistent with the higher aims of the enterprise.

Connect strategy to execution. The next step is to make sure everyone in the SAFe Portfolio aligns their work to the most important things to the enterprise. Strategic themes explicitly translate business strategies into tangible guidance that aligns the portfolio vision, lean budgets, and epics to enterprise priorities and subsequently inform the work of teams and trains in the portfolio.

Speak with a common language. It's difficult to achieve alignment if there's inconsistency in how the organization describes important roles, processes, events, and artifacts. SAFe provides a common language and promotes practices (backlogs, ART boards, solution intent, portfolio vision, and so on) that maintain a common view of the work and the resulting solutions.

Constantly check for understanding. Creating alignment requires regular reinforcement. SAFe events (iteration planning, backlog refinement, PI planning, ART syncs, portfolio syncs) and SAFe artifacts (backlogs, team boards, ART boards, portfolio canvas) are just some of the tools that help the SAFe organization stay aligned. Face-to-face conversations are also essential for checking for understanding.

Understand your customer. SAFe promotes continuous exploration with customer centricity and design thinking to gather inputs and perspectives from diverse stakeholders and information sources to ensure that the items in the

backlogs are aligned with the most important voice of all—the customer.

Transparency

Solution development is complex. Often, things go wrong or do not work out as planned. Without openness, facts are obscure, and decision-making is based on speculative assumptions and a lack of data. No one can fix a secret.

To ensure transparency—*trust* is needed. Trust exists when everyone can confidently rely on one another to act with integrity, particularly in times of difficulty. Without trust, it is impossible to build high-performing teams and trains or build (or rebuild) the confidence needed to make and meet reasonable commitments. Trust-based environments are also fun and motivating. Simply put, the new way of working promoted by SAFe will struggle to succeed without a culture of transparency and trust.

The following actions can help build a culture of transparency and trust in the SAFe enterprise.

Create a trust-based environment. Trust requires action, not just a feeling. People at every level of the organization must be willing to trust others and be trustworthy themselves. It means making and keeping commitments. It also means relinquishing control and trusting others to make and keep their commitments.

Communicate directly, openly, and honestly. A frequently used mantra in SAFe is “the facts are friendly.” Problems cannot be solved if they are hidden. In a trust-based environment, information is shared without embellishment or blame to resolve issues as quickly and effectively as possible.

Turn mistakes into learning moments. People often learn more from their mistakes than from their successes, but this is effective only when those mistakes can be acknowledged without fear of retribution or punishment. Address mistakes as “learning moments” [2] to create the psychological safety needed to quickly surface and resolve errors.

Visualize work. Making all work visible is essential to transparency. In SAFe, this begins with all work at every level being captured in a continuously refined backlog. Other tools such as Kanban boards, ART boards, PI objectives, solution intent, collaboration tools, and shared knowledge repositories support the aim of keeping work visible and accessible to all.

Provide ready access to needed information. Information that is difficult to find has the same impact as if that knowledge were intentionally hidden. True

transparency requires that information is easily accessible to all who need it and that the location and means of access are well known. It requires a willingness to help each other find required information when the location is unclear and relentlessly improve systems to make accessing information as frictionless as possible.

Respect for People



First we build people, then we build cars.

Ñ Fujio Cho, former Chairman, Toyota

A Lean-Agile approach doesn't implement itself or perform any real work. In reality, people do all the work, and people receive all the value from the work. Since people are the focal point of how enterprises create value with SAFe, respect for people must be considered in every aspect of the new way of working.

Respect is a basic human need. When treated with respect, people are unleashed to evolve their practices and contribute their creativity. Conversely, people cannot commit to another person, their teams, or their organizations if they feel a lack of respect. When disrespect is widespread and tolerated, it creates a toxic work environment, poor performance, and a high attrition rate [3].

The following suggestions provide just a few ways to cultivate a culture of respect for people in an organization.

Hold precious what it is to be human. This is the literal meaning of "respect for people" described in Lean. Elaborated further, it means fostering a corporate culture that enhances individual creativity and values teamwork while honoring mutual trust and respect [4].

Value diversity of people and opinions. Another way to show respect is to build organizations that include people with various personal and professional backgrounds. However, just hiring a diverse workforce is insufficient. Respect for people requires listening to and valuing perspectives and viewpoints different from our own.

Grow people through coaching and mentoring. Respect for people goes beyond a fundamental moral obligation. The practical business implication is that "it is necessary to develop good people in order to make good products." One way to help people grow is to facilitate connections with others inside and outside

the organization who can contribute to each person's development journey.

Embrace Øyour customer is whoever consumes your work. Lean and Agile methods are customer-centric, as both recognize that customers are the ultimate beneficiaries of value. In addition, Lean explicitly addresses the fact that all who consume your work, including people inside the organization, are customers too. Treating customers with respect and empathy produces products and services that address their real problems.

Build long-term partnerships based on mutual benefit. Many suppliers are required when building the world's most complex systems. When applied to suppliers, respect means holding them in the same high regard as customers, challenging them, and helping them improve. This is done not by bullying or pressuring tactics but by creating long-term relationships defined by Øwin-win contracts based on mutual benefit and accountability.

Relentless Improvement

The relentless pursuit of perfection has always been one of the core tenets of Lean. While unattainable, striving for perfection leads to continuous improvements to products and services. In the process, companies have created more and better products for less money and with happier customers, leading to higher revenues and greater profitability.

But improvement requires learning. Rarely are the causes and solutions for problems that organizations face clear and easily identified. Relentless improvement is built on a series of small iterative and incremental improvements and experiments that enable the organization to learn its way to the most promising answer to a problem.

The following moves can help build a culture of relentless improvement in a SAFe enterprise:

Create a constant sense of urgency. Improvement activities are essential to the survival of an organization and should be given priority, visibility, and resources. They require an intense focus on delivering value to customers by providing products and services that solve their problems in a preferred way over the organization's competitors. Organizations that become complacent and fail to relentlessly improve with urgency risk losing customers and may ultimately go out of business.

Build a problem-solving culture. Problem-solving is the driver for relentless improvement. It recognizes that a gap exists between the current and desired

states, requiring an iterative process to achieve the target state. Iterative Plan-Do-Check-Adjust (PDCA) cycles provide the process for iterative problem-solving on small adjustments as well as breakthrough innovations. The goal is to have a culture of everyone improving all the time.¹

Reflect and adapt frequently. It's vital in SAFe to periodically pause from the never-ending backlog of new work to openly identify and address shortcomings of the process at all levels. Improvements should be managed and prioritized just like any other story or feature because improvement requires real work and consumes real capacity of teams and trains, as well as for those who guide the portfolio processes.

Let facts guide improvements. Improvements based on opinion or conjecture will likely focus on symptoms instead of true root causes. Improvement results are objectively measured, focusing on empirical evidence. This helps an organization concentrate more on the work needed to solve problems and less on assigning blame or on pursuing improvements that are not solving the original problem.

Provide time and space for innovation. Improvements should be designed to increase the effectiveness of the entire system that produces the sustainable flow of value instead of optimizing individual teams, silos, or subsystems. Everyone at all levels should embrace improvement thinking, but improvements in one area, team, or domain should not be made to the detriment of the overall system. SAFe Principle #2 [Apply systems thinking](#) expands on this idea further.

Leadership is Required

Consistently applying the SAFe core values requires the active support of [Lean-Agile Leadership](#) and a [Continuous Learning Culture](#). Leaders in a SAFe organization should exemplify the core values combined with the Lean-Agile Mindset, SAFe [Principles](#) and practices, and an orientation toward creating value for customers. In turn, modeling these mindsets and behaviors creates a persistent and meaningful values-based culture for teams and their stakeholders.

Learn More

[1] Labovitz, George H., and Victor Rosansky. *The Power of Alignment: How Great Companies Stay Centered and Accomplish Extraordinary Things*. Wiley, 1997.

[2] Ridge, Gary. <https://thelearningmoment.net/welcome-to-learning-moment/>

[3] Hu-Chan, M. *The basic human need some companies fail to meet.* Inc.com. <https://www.inc.com/maya-hu-chan/the-basic-human-need-some-companies-fail-to-meet.html>

[4] Miller, J. *Exploring the ØRespect for PeopleØ Principle of the Toyota Way.* Gemba Academy, 2017. https://blog.gembaacademy.com/2008/02/03/exploring_the_respect_for_people_principle_of_the/

[5] Liker, Jeffery K. *Developing Lean Leaders at All Levels.* Lean Leadership Institute Publications, 2014.

Last update: 13 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



It is said that improvement is eternal and infinite. It should be the duty of those working with Kanban to keep improving it with creativity and resourcefulness without allowing it to become fixed at any stage.

—Taiichi Ohno

[Adjust Cookie Settings](#)

SAFe Team Kanban

SAFe Team Kanban is an Agile method used by teams within an ART to continuously deliver value. SAFe Kanban teams apply a flow-based process to their daily work and operate within the ART iteration cadence.

Kanban teams apply a flow-based process to their daily work and operate within the [Agile Release Train \(ART\)](#) iteration cadence. The Kanban method provides a strategy for optimizing the flow of value using a visual, pull-based system instead of work being pushed to or by the team. ÓKanban comprises the following three practices working in tandem [1]:

- Defining and visualizing a workflow
- Actively managing items in a workflow
- Improving a workflow.Ó

In SAFe, Kanban systems manage the backlog and flow of work at every level of the Framework. Each reflects a teamÓs unique process for delivering value and its current workflow and capacity.

Details

Most [Agile Teams](#) use [SAFe Scrum](#) as their primary method to deliver value. However, some teams have a rapid and uneven arrival of work and fast-changing priorities that lowers the value of the time investment in iteration planning. In these cases, teams often choose SAFe Team Kanban instead. For example, [System Teams](#), operations, support, hardware, and various business teams often find SAFe Team Kanban a good choice for their context.

In addition, the level of visibility and flow that Kanban provides causes it to spread to different parts of the organization. Today, many organizations adopt Kanban to help embrace [Lean-Agile Principles](#) across all aspects of business, from marketing to finance, human resources to legal, security to compliance, operations to Agile Teams, and more.

Find a Course:

Go

Like all SAFe Agile Teams, Kanban teams determine how they manage their work. They create and refine backlog items—typically expressed as [Stories](#) with acceptance criteria—to define and achieve their Team [PI Objectives](#). They then build, integrate, test, validate, and deploy the new functionality, ensuring [Built-in Quality](#).

Since Kanban teams typically have all the roles and skills needed to develop and deliver increments of value, they operate with the minimum possible constraints and dependencies with other teams. A self-managed and cross-functional Kanban team creates a more enjoyable, fun, and productive work environment with constant communication, constructive conflict, and dynamic interaction.

The SAFe Team Kanban Board

A Kanban system includes a “Kanban board” used to visualize and manage the work flowing through the system. The standard elements of a Kanban board are shown in Figure 1.

Figure 1. Elements of a Kanban board

Figure 1. Elements of a Kanban board

Work in Process (WIP) limits set the maximum number of items for an individual workflow state.

Columns represent a series of steps, each representing an activity that collectively defines the team's workflow.

Cards represent work items, such as user stories and enablers.

Swim lanes group and highlight related work items to define the team's workflow. Typical use of swim lanes includes separating work for different classes of service, such as lanes for expediting, individual workflows, cross-team dependencies, features, and more.

Policies specify how work is managed, such as exit or entry criteria for moving a work item from one state to another or defining the rules for service classes.

See Figure 4 for an example of a more fully elaborated Team Kanban system with columns (steps) used by an actual team.

The SAFe Team Kanban Method

While Kanban guides managing work in a flow-based system, it is not explicit concerning the roles, responsibilities, and events that teams use to apply Kanban as their Agile practice. SAFe addresses this as illustrated in Figure 2 below. Each element of the SAFe Team Kanban method is described in the following sections.

Figure 2. SAFe Team Kanban method overview

Figure 2. SAFe Team Kanban method overview

Team Backlog

The [Team Backlog](#) contains all the upcoming work needed to advance the solution.

Teams continually refine the backlog to ensure it has some stories ready for implementation without significant risk or surprise.

During [PI Planning](#), teams decompose [Features](#) into stories in the backlog and establish their [PI Objectives](#). The team's local concerns (other new functionality, defects, refactors, tech debt, and maintenance) are also in the backlog. These stories reload the team backlog for the upcoming PI. But since PI planning is high level, teams will likely need to adjust their plans as stories are refined, acceptance criteria are established, and other new facts emerge. Moreover, feedback from previous increments, the [System Demo](#), and other groups with whom they collaborate provide rolling wave updates to the backlog and flow of work.

Plan

Although the flow of work is continuous, planning is valued in Agile, and Kanban teams are no exception. Many Kanban teams plan weekly to coordinate their work, replenish stories in the backlog, and address dependencies and fixed date commitments. Some Kanban teams find it convenient to align their weekly planning with the iteration cadence of the ART. Once complete, the team records the planned work in a visible place, such as a physical Kanban board or Agile project management tooling. A weekly planning time box of 60-90 minutes is typical.

As part of planning, SAFe Kanban teams often establish [Iteration goals](#), which provide [Agile Teams](#), Agile Release Train (ART) stakeholders, and management with a shared language for maintaining alignment, managing dependencies, and making necessary adjustments during the execution of the [PI](#).

Kanban teams continually replenish the backlog and identify any stories that need to be completed in a specific timebox (fixed date or expedited class of service). They also ensure a sufficient amount of prioritized backlog items exists for at least one or two iterations. That's their plan and commitment to the business.

Deliver

In SAFe, teams apply Kanban within the development cadence and synchronization requirements of the [Agile Release Train \(ART\)](#). They, like others, [Release on Demand](#). This cadence and synchronization facilitate alignment, dependency management, and fast integrated learning cycles ([SAFe Principle #4](#)).

The Kanban system visualizes all active and pending work, workflow states, and WIP limits. A work item can be pulled into a state only when the number of items currently in that step is below the WIP limit. A few activities (typically beginning and end) may not be limited. WIP limits are defined and adjusted by the team, allowing it to adapt quickly to the variations in the flow of complex system development.

Team Sync

In addition to the weekly planning meeting, Kanban teams coordinate their work throughout the week. They decide if these syncs are cadence-based or ad hoc. The tempo and timing can vary significantly based on stages of development. A typical pattern is holding a team sync weekly around midweek. Kanban teams typically discuss the following kinds of topics during this time:

- Review how work is flowing and remove impediments
- Peer review of WIP and adjust upcoming planned work
- Review and accept stories
- Discuss improvements to the team's process
- Planning for the system demos that occur throughout the PI

Monitoring fix-date commitments and flow metrics

In a flow-based system, the team can release work into later stages without formal signoffs or approvals, subject to the team's policies. Therefore pairing and swarming are routine and informal to help better ensure built-in quality before deployment.

System Demos

Like all SAFe Agile Teams, Kanban teams participate in the ART's system demos, representing another form of syncing within the team and across the ART. This synchronization ensures the integration of the team's work into the solution, including demoing progress. It also fosters collaboration with other groups and stakeholders to assess the solution, making mid-course corrections as necessary.

Increment

Kanban teams deliver small increments of value throughout the PI, representing how new functionality evolves. Each increment is additive and is a working, tested, and functional solution element.

Retrospective

Kanban teams periodically reflect and identify new ideas to improve their process. These improvements often result in updates to the Kanban board to capture the revised process. Retrospectives help instill the concept of relentless improvement— one of the SAFe [Core Values](#), ensuring that the team continually improves. While optional, teams may conduct a retrospective, every iteration aligned with the ART or at least once per PI, typically just before the ART's [Inspect & Adapt \(I&A\)](#) event. In this way, the knowledge from their team retrospective can inform the problem-solving part of the I&A.

Kanban Roles

While Kanban is generally less specific on team roles, SAFe applies the two Scrum special team roles—the Product Owner and the Scrum Master/Team Coach. These roles have emerged in practice as equally helpful to Agile Teams applying SAFe Team Kanban (Figure 3).

A diagram showing the SAFe Team Kanban system. At the top, there is a box labeled "Product Owner". Below it, a horizontal line connects to a box labeled "Team Coach". This line then extends downwards to a larger box labeled "Agile Team".

Figure 3. Product Owner and Team Coach roles in SAFe Team Kanban

As with Scrum, the [Product Owner](#) is a member of the Agile Team responsible for maximizing the value delivered by the team and ensuring that the Team Backlog is aligned with customer and stakeholder needs. As a member of the extended [Product Management](#) function, the PO is the team's primary customer advocate and link to business and technology strategy. This role enables the team to balance the needs of multiple stakeholders while continuously evolving the Solution. They serve as the Kanban process intake owner and prioritize the team backlog to help ensure the team is building the right thing.

The SAFe [Scrum Master/Team Coach](#) (SM/TC) is a servant leader and coach for an Agile Team. They help educate the team in Kanban, [Built-in-Quality](#), and [SAFe Scrum](#), ensuring that the agreed Agile processes are followed. They also help remove impediments and foster an environment for high-performing team dynamics, continuous flow, and relentless improvement.

Establishing the Team Kanban

Effective Kanban systems are established based on the needs of each Agile Team and the type of work performed (for example, software development, hardware, marketing). Establishing the Kanban system typically involves the entire Agile Team with the guidance and facilitation of an experienced coach. The SAFe extended guidance article, [Applying Kanban in SAFe](#), describes establishing a Kanban system and how they connect in SAFe. Figure 4 illustrates an example of a more fully elaborated Team Kanban system.

Figure 4. Example team Kanban system

Figure 4. An example team Kanban system

Improving and Measuring Flow

Measuring Flow

Kanban systems provide a rich data set that can identify bottlenecks and improve flow. Several standard metrics can measure different aspects of flow. These include Flow Distribution, Flow Velocity, Flow Time, Flow Load, Flow Efficiency, and Flow Predictability. See the [Measure and Grow](#) and [Team Flow](#) articles for information on measuring flow.

Optimizing Flow

A team's Kanban board should evolve iteratively and continuously adapt to fit the team's needs. After defining the initial process and WIP limits and running the system, bottlenecks should become visible and can be addressed. Other changes to optimize flow might include adding, merging, or splitting steps, adding buffers, or

redefining workflow states.

Estimating Work

Due to the rapidly changing nature of work items, there is typically less emphasis on estimating stories than in Scrum. Instead, Kanban teams look at the work needed, right-size it by splitting large items where necessary, and pull the resulting work through the Kanban system to completion.

However, SAFe teams still estimate the demand for work against their capacity during PI planning and help contribute to estimates for cross-team backlog items (for example, features and epics).

Learn More

[1] The Kanban Guide. <https://prokanban.org/the-kanban-guide/>

[2] Anderson, David J. *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, Washington: Blue Hole Press, 2010.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

PARTNER

[Becoming a Partner](#)

CONTENT &
TRADEMARKS

SCALED AGILE,
INC

[Posters & Graphics](#)

[Watch and download](#)

[SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



A Scrum Master is like an orchestra conductor, guiding a group of individuals to create something that no one of them could create alone.

~Mike Cohn

SAFe Scrum Master/Team Coach

[Adjust Cookie Settings](#)

Note: For more on SAFe Scrum, please read the additional Framework

articles in the Scrum series, including [SAFe Scrum](#), [Iterations](#), [Iteration Planning](#), [Iteration Goals](#), [Iteration Review](#), and [Iteration Retrospective](#)

The SAFe Scrum Master/Team Coach (SM/TC) is a servant leader and coach for an Agile team who facilitates team events and processes, and supports teams and ARTs in delivering value.

They help educate the team in Scrum, Built-in-Quality, Kanban, and SAFe and ensure that the agreed Agile processes are followed. They also help remove impediments and foster an environment for high-performing team dynamics, continuous flow, and relentless improvement.

Scrum Master Stories: Yolanda

Find a Course:

Details

In SAFe, the Scrum Master/Team Coach (SM/TC) assists the team in meeting their delivery goals. They coach teams in self-organization and self-management and help them coordinate and participate in [Agile Release Trains \(ARTs\)](#) events, increasing the effectiveness of SAFe across the organization.

SAFe SM/TC are integral members of an [Agile Team](#) and share responsibilities with the team for their overall performance. The SM/TC has the specialty skills that support adopting [SAFe Scrum](#) practices, ensuring no substantial gaps, and the team knows how to plan, execute, review and retrospect. In addition, SM/TCs can actively coach [SAFe Team Kanban](#) teams and help each Agile Team achieve [Team Flow](#).

Characteristics of a SAFe Scrum Master/Team Coach

The SM/TC role is a team member who has the responsibility to help the team achieve its goals. They do this by teaching and coaching SAFe Scrum and SAFe Team Kanban, and by supporting SAFe principles and practices. They also help identify and eliminate bottlenecks to flow.

SM/TCs come from various backgrounds and roles, and they are in high demand. Although they are not typically people managers, SM/TCs are influential members of an Agile Team, and they should have the following attributes:

Empathetic ☰ Support the team by displaying an authentic understanding and concern for a team member's beliefs or feelings. In turn, the team is more likely to build relationships with others, resulting in higher levels of collaboration and performance. Empathy is a crucial ingredient of trust, which is essential for people to accept and welcome coaching.

Conflict navigator ☰ Supports team members in resolving interpersonal conflicts, problem-solving, and decision-making. Agile coach and author Lyssa Atkins opines, "Navigating conflict is our new mindset, in which we help teams move from conflict to constructive disagreement as a catapult to high performance." [2]

Servant leader ☰ Persuades rather than uses authority. As servant leaders, SM/TCs focus on the needs of team members and those they serve, intending to achieve results aligned with the organization's values, principles, and business objectives. [3] They have choices in how they collaborate with the team depending on the situation and their accountability for team performance. SM/TCs should have options for achieving their responsibilities. For example, when it comes to events their accountability should be "ensuring that all team events take place and are positive, productive, and kept within the timebox." SM/TCs can facilitate the events or let the team self-manage and facilitate their own events. Rotating the responsibilities for facilitating events and meetings is essential to the team's growth and its ability to self-manage.

Mentor ☰ Supports the personal development of team members, helping them gain a continuous learning mindset. They guide the team to find solutions to their problems independently instead of being given the answers.

Transparent ☰ Transparency is a [Core Value](#) of SAFe and one of the pillars of empiricism. The SM/TC is open to feedback and appreciates transparency from others. They help the team provide transparency by ensuring artifacts are inspected, identifying significant differences between expected and actual results, and detecting anti-patterns.

Coach ☰ The SM/TC understands and educates the team on methods beyond Scrum, such as SAFe, Kanban, Flow, Built-in Quality, and more. They often have advanced training and experience in one or more technical and business domains.

Responsibilities

The SM/TC fulfills many critical responsibilities in performing the role, as illustrated in Figure 1. Each of these responsibilities is described in the sections that follow.

Figure 1. SAFe SM/TC Responsibilities

Figure 1. SAFe SM/TC responsibility areas

Facilitating PI Planning

SM/TCs play an essential role in [PI Planning](#). They collaborate with other SM/TCs and the [Release Train Engineer \(RTE\)](#), working actively with the team during PI planning. An effective SM/TC is critical to a successful event, and they typically do the following activities to help facilitate PI planning:

Prepare for PI Planning Before the event, the SM/TC ensures the team is briefed on upcoming features by [Product Managers](#), [Business Owners](#), and other stakeholders, as illustrated in Figure 2. They help Agile Teams and the [Product Owner](#) identify local stories, maintenance, defects, tech debt, and other work the team needs to accomplish during the upcoming PI.

Figure 2. Preparing for PI planning

Figure 2. Preparing for PI planning

Draft PI plans The SM/TC facilitates the team in creating a draft PI plan for the PI's iterations, writing draft **PI Objectives**, and identifying ART risks and issues. The SM/TC also helps the team set up their digital or physical planning areas, providing visual radiators that create transparency and collaboration. They help the team determine their capacity and keep within this constraint.

Coordinate with other teams SM/TCs help ensure cooperation and communication during the event. During PI planning, they usually secure subject matter experts (SMEs) and ART stakeholders and foster communication with other teams to determine how they will collaborate on feature development and resolve dependencies.

Create team PI objectives SM/TCs help teams create team PI objectives, the things they intend to accomplish in the upcoming PI. They ensure the objectives are written before the draft plan review and that a proper mix of committed and uncommitted goals is present.

Review final plans and business value Before the final review, SM/TCs help ensure PI objectives are Smart (Specific, Measurable, Achievable, Realistic, and Time-bound) and are written in a way everyone can understand. The SM/TC often

facilitates Business Owner and team collaboration during business value assignments.

Supporting Iteration Execution

SM/TCs support Agile Teams during the iteration, increasing the likelihood of achieving its iteration goals and PI objectives. For example they:

Facilitate team events ☰ Agile Teams use cadence-based events to coordinate and sync their efforts. While Scrum and Team Kanban operate somewhat differently, all teams need to plan, sync, review, inspect their work, and hold retrospectives. Figure 3 illustrates the events (or activities) that Agile Teams typically do during an iteration.

Figure 3. SM/TCs' role in ensuring successful team events

Figure 3. SM/TCs' role in ensuring successful team events

Work within the ART's cadence ☰ SMs/TCs help teams apply Scrum or Kanban within the development cadence and synchronization requirements of the [Agile Release Train \(ART\)](#). This cadence and synchronization facilitate alignment, dependency management, Release on Demand, and fast integrated learning cycles

(SAFe Principle #4).

Collaborate with the PO Since the Product Owner (PO) is accountable for maximizing the solution's value resulting from the team's work, an essential aspect of the SM/TC's role is supporting the PO. They do this by:

- Helping the team understand and apply the tools and techniques for [Customer-Centricity](#) and [Design Thinking](#) to build the right thing at the right time
- Ensuring the team understands the need for clear and concise team backlog items and aligns to the ART's capacity allocation for each work item type.
- Helping the team apply empirical planning and development where progress is evaluated based on observation and experimentation of working solutions in small increments
- Facilitating stakeholder collaboration as requested or needed

Improving Flow

SM/TC can significantly improve the team's flow of work, eliminating bottlenecks, delays, and waste. This coaching often includes the following activities:

Establish the team Kanban board SAFe teams use a Kanban board to visualize their work and enhance flow. Implementing an effective Kanban system adapted to meet the needs of a specific Agile Team is based on the type of work performed for example., marketing, software development, hardware), the team members' skills, and their role in the ART. Creating the Kanban system is best done by involving the entire Agile Team with the guidance and facilitation of an experienced coach like the SM/TC. The SAFe extended guidance article, [Applying Kanban in SAFe](#), describes how to establish a Kanban system and how the Kanban systems are connected in SAFe. Figure 4 illustrates an example of a Team Kanban board.

Figure 4. Example Team Kanban board

Figure 4. Example Team Kanban board

Measure and optimize flow ☰ SM/TCs help the team establish metrics to assess and improve its overall performance. Specific measures for *flow*, *competency*, and *Outcomes* are described in [Measure and Grow](#). Flow Metrics help the SM/TC and the team evolve its process iteratively and continuously adapt to fit the team’s needs. After defining the initial process and WIP limits and executing for a while, bottlenecks should become visible. If not, the team refines the process or further reduces some WIP limits until it becomes evident that a workflow state is overloaded or starving. Other coaching opportunities for optimizing flow might include merging or splitting steps, adding buffers, swim lanes, and classes of service, or redefining workflow states.

Build quality in ☰ Agile Teams operate in a fast, flow-based system to develop and release high-quality business capabilities quickly. The SM/TC helps achieve this by coaching [Built-in quality](#) practices, which enable fast, reliable execution and helps ensure that needed and frequent changes are made efficiently and effectively.

Building High-Performing Teams

Creating healthy Agile Teams is essential to creating high-value increments of working solutions. Fortunately, many of the ingredients for high-performing teams are built into SAFe by design. For example, Agile Teams in SAFe are small, cross-functional, and self-organizing. They are empowered to define and execute the work needed to accomplish the team's objectives and those of the ART. Everyone agrees that all increments should meet a shared, scalable definition of done. SM/TCs play a critical role in building high-performing teams and accomplish this through the following types of activities:

Foster and support Agile Team attributes While every team is different, there are common characteristics that high-performing teams share. SM/TCs are responsible for supporting and fostering the following Agile Team attributes:

- Self-management and taking ownership and accountability
- Aligned and collaborative
- Success focused on clear goals and purpose
- Influential decision-makers who understand their work's impact on others
- Operate with open and transparent communication and trust
- Value diversity and healthy conflict
- Provide effective, timely feedback
- Highly engaged and have fun with work, and each other

Encourage high-performing team dynamics SM/TCs foster an environment for high-performing team dynamics, continuous flow, and relentless improvement. The SM/TC mentors the team and creates an atmosphere of mutual respect, helping resolve interpersonal conflicts, and identifying growth opportunities. They assist the team in *focusing* on creating increments of high value for each iteration.

Become a more effective Scrum Master/Team Coach Every servant leader knows that their growth comes from facilitating the development of others who deliver the results. SM/TCs serve the team and the larger organization. The SM/TC supports the overall adoption of SAFe across the enterprise by coaching stakeholders and non-agile teams on effective interactions with Agile Teams, participating in the SM/TC Community of Practice, and supporting the organization's [SAFe Practice Consultants \(SPC\)](#).

Serve as Lean-Agile Leaders SM/TCs also advance the adoption of SAFe. They lead by example and incorporate the [Lean-Agile Mindset](#) and [SAFe Lean-Agile Principles](#). They integrate these concepts into their responsibilities and serve as a role model for others to follow.

Foster collaboration on the team The SM/TC role fosters more effective and cohesive teams, enabling better business outcomes, solutions, and products. They offer observations, feedback, guidance, and advice based on what they know and have seen work.

Coach with powerful questions However, SM/TCs do not have all the answers. Instead, they can ask *powerful questions* to uncover what's essential, then guide others to tap into their knowledge and expertise. Some examples of powerful questions include:

What brings us to this inquiry?
What other possibilities or options exist?
What is it we're not seeing?
What do we need to do to reach a deeper level of understanding?
If success was guaranteed, what actions would you take?

By asking powerful questions, SM/TCs help teams improve their performance, solve problems more independently, make better decisions, learn new skills, and better reach their goals.

Resolve team conflicts Teamwork is the ultimate competitive advantage.

However, many teams are dysfunctional, according to Patrick Lencioni, consultant and author of *Five Dysfunctions of a Team*. In his book, Lencioni suggests that an absence of trust leads to the other four dysfunctions. SM/TCs helps address these five dysfunctions with the SAFe practices, as illustrated in Figure 5.

Figure 5. The SAFe SM/TC helps address the five dysfunctions of a team

Figure 5. The SAFe SM/TC helps address the five dysfunctions of a team

Develop team skillsets SM/TCs work with team members and their functional

managers to help them acquire T-shaped skills. A *T-shaped* individual has broad, general expertise in many areas and is an expert in one of these disciplines. SM/TCs encourage team members to pair with others to expand their skills, take on tasks in another discipline and business domain, and participate in training courses and reading books to become continuous learners.

Improving ART Performance

SM/TCs help Agile Teams improve the overall ART performance through the following activities:

Facilitate cross-team collaboration Cross-team collaboration is a hallmark of high-performing teams. Agile Teams cooperate across departments to bring whole product solutions to market. SAFe SM/TCs nurture an environment where cross-team collaboration thrives and is supported by practices that offer opportunities for teams to work together, for example:

- Alignment to ART PI objectives, Vision, and Strategic Themes during PI planning and addressing dependencies using the ART board
- Representing the team in the Coach Sync, PO, and ART Syncs
- Attending other team's events and demos with relevant team members
- Participating in the ART's [System Demos](#) and [Inspect & Adapt](#) events

One of the significant benefits of working on and across teams is colleagues learn from one another. On an Agile Team, learning new skills makes everyone more valuable to the organization and better equipped to support each other's work. It also guards against specialty skills becoming a bottleneck, which increases delays and reduces quality.

Build trust with stakeholders The SM/TC helps the team build trust. SAFe relies on a rolling wave of short-term commitments from Agile Teams and ARTs to assist with business planning and outcomes, resulting in improved alignment and trust between development and business stakeholders. While solution development is uncertain by its very nature, the business depends on teams for some amount of reliable, predictable forecasting. Too little predictability and the company can't plan. Too much, and the organization has committed to longer-term plans, which are unreliable and limit agility. Business and technology stakeholders need something in between, which is the primary purpose of PI objectives.

Coach the IP Iteration SM/TCs help ensure the team does not schedule any work for the IP Iteration during PI planning. Instead, they coach teams to use this iteration as an estimating buffer for meeting PI objectives and providing dedicated time for innovation, continuing education, PI planning, and the Inspect and Adapt (I&A) events.

Help the team inspect & adapt Ensures the team is prepared for the Inspect &

Adapt event, including the PI System Demo, quantitative and qualitative measurement, and the retrospective and problem-solving workshop. They help guide the team in the I&A activities and stay within the allotted timeboxes.

Facilitate the problem-solving workshop SM/TCs coach teams in root cause analysis, the Five whys and Pareto analysis. They ensure that the relevant work needed to deliver the identified improvements is planned and added to the Team Backlog.

Full or Part-Time Role?

The SM/TC can be a part-time or full-time role, depending on the size of the team, the context, and other responsibilities. However, it can be challenging for an Enterprise to justify the need for a full-time SM/TC for each Agile Team. SAFe takes a pragmatic approach, where sometimes a team member assumes the role along with other duties, or an accomplished SM/TC can support more than one team. However, during initial SAFe adoption, the job can be more intensive. It's often beneficial to hire external SM/TC consultants to mentor the teams and help them become experienced in their roles and SAFe. These consultants will work with multiple teams and new SM/TCs. And, of course, adequate training and experience are required to be effective.

Learn More

[1] Designing powerful questions. <https://conversational-leadership.net/powerful-questions/>

[2] Scrum Master as conflict navigator. <https://medium.com/serious-scrum/scrum-master-as-conflict-navigator-de5c6a162fe>

[3] The Scrum Master as a servant leader.

https://www.scrum.org/resources/blog/scrum-master-servant-leader#_ftnref7

Last Update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Business people and developers must work together daily throughout the project.

~ Agile Manifesto

Product Owner

The Product Owner (PO) is the Agile team member primarily responsible for maximizing the value delivered by the team by ensuring that the team backlog is aligned with customer and stakeholder needs.

As a member of the extended [Product Management](#) function, the PO is the team's primary customer advocate and primary link to business and technology strategy. This enables the team to balance the needs of multiple stakeholders while continuously evolving the Solution.

Details

For most enterprises moving to Agile, this is a new—and typically full-time—role for each [Agile Team](#). Each PO represents the needs of customers and the business within a particular Solution domain, typically co-represented by a Product Manager. Together, they ensure that product strategy and implementation remain connected throughout the value stream.

Serving as the “voice of the customer” for the team entails a broad range of responsibilities. The PO must build and manage key relationships, synthesize information from multiple sources, maintain business alignment in the [Team Backlog](#), and communicate effectively with various audiences—all with a bias toward delivering and learning quickly.

Responsibilities of the Product Owner

The PO's responsibilities can generally be categorized into five primary areas, as shown in Figure 1.

Figure 1. Product Owner areas of responsibility

Figure 1. Product Owner areas of responsibility

Each of these areas of responsibility is described in the sections below.

Connecting with the Customer

Ensuring that ARTs are continually building the right things and building them right is a never-ending process. Product strategy, design, and implementation must evolve with ever-changing customer desires and business needs. The PO, in close partnership with Product Management, applies a customer-centric mindset along with design thinking tools to guide the ART toward delivering solutions that are desirable, viable, feasible, and sustainable. The PO applies [Customer Centricity](#) and [Design Thinking](#) in the following ways:

Know the customer ☰ Value is determined by the customer; therefore, the PO is

keenly aware of the needs of the people to whom their products are delivered. Customers may be internal or external to the enterprise and may have direct or indirect relationships with the PO. Whether they consume products, services, systems, APIs, platforms, or other solutions, customers' wants, needs, and preferences are continually explored by the PO.

Know the stakeholders ☰ Product design and implementation must also reflect the needs of non-customer stakeholders. [Business Owners](#), [Lean Portfolio Management](#), Product Management, [System Architects](#), and fellow POs, for example, rely on the cadence and quality of the team's output. The PO identifies key stakeholders and balances their needs with those of the customer.

Identify the problem to be solved ☰ Good products solve specific problems. What's more, they solve specific problems that are *worth* solving. Identifying problems that customers want to be solved is the first element of design thinking. In this context, the PO discovers a range of customer needs through divergent thinking tools, then identifies the 'jobs to be done' that are most worth pursuing.

Develop whole-product solutions ☰ Solutions that address a range of customer needs are more valuable than those that target a single need. POs aim to deliver whole-product solutions by understanding the desired customer experience, guiding the development of candidate designs through the [Lean UX](#) process, and delivering tested concepts that maximize customer satisfaction and loyalty.

Contributing to the Vision and Roadmap

While product managers contemplate the solutions and experiences an ART *should* deliver, POs understand what solutions and experiences the ART *can* deliver. This practical insight is a valuable contribution to the vision and roadmaps that guide solution implementation. The PO applies this pragmatic insight in the following ways:

Understand market forces ☰ Market rhythms, market events, sudden opportunities, competitive threats, and changing regulations significantly influence product strategy. POs regularly engage with Product Management to analyze market research and understand the business drivers that trigger Feature requests.

Represent the end user ☰ Through frequent interviews, gemba, and reporting, POs are strongly connected to the needs and experiences of their products' end users. Objective insights about how end-users interact with solutions and the features they desire most ensure that the vision and roadmap contain real value.

Assist with ART Backlog prioritization In collaboration with Product Management, System Architects, Release Train Engineers (RTEs), and other stakeholders, POs guide the sequencing of features over time toward the best economic outcomes. Through their understanding of which problems need to be solved, which solutions would best solve them, and the feasibility of delivering those solutions, POs help ensure that the vision and roadmap are reflected in the [ART Backlog](#).

Educate the ART during PI Planning The [Vision](#) and [Roadmap](#) are living artifacts created and adjusted in alignment with business and technical strategy, but a portion of them is always in scope for implementation. The PO assists with communicating the vision and roadmap during PI Planning to ensure teams are aligned and ready to execute against them.

Managing and Prioritizing the Team Backlog

With input from Product Management, System Architecture, and other stakeholders, the PO is primarily responsible for maintaining the content and the conceptual and technical integrity of the team backlog. Consisting of user stories, enablers, and defects, the backlog must always contain work ready to be pulled for implementation by the team and aligned with the most current needs of customers and stakeholders. The PO manages the ongoing integrity of the team backlog through the following activities:

Guide Story creation While any team member can write stories at any time, it is the PO's responsibility to ensure that they are well-formed and aligned with product strategy. The PO clarifies story details, applies user-story voice, ensures "INVEST" characteristics are present, assists with story splitting, defines enablers, and incorporates behavior-driven design (BDD) to ensure stories support continuous value flow. The PO also allows space for "local" stories and spikes that advance product design but are not derived explicitly from ART-level features.

Prioritize backlog items Achieving continuous value flow requires that the highest-value backlog items are delivered in the shortest sustainable lead time and in the right sequence. The PO enables this by regularly ordering backlog items according to their cost of delay and communicating that sequence to the team during backlog refinement and [Iteration Planning](#).

Accept Stories The PO works with the team to agree on accepted story completion. This includes validating that the story meets acceptance criteria, that it has the appropriate, persistent acceptance tests, and that it otherwise complies with its Definition of Done (DoD). In so doing, the PO assures that quality is built in.

Support Architectural Runway POs do not typically drive technological decisions, but they make space in the backlog to support the implementation of [Architectural Runway](#). They collaborate with System Architects to craft enablers and work with stakeholders to establish appropriate capacity allocations.

Supporting the Team in Delivering Value

Value is created when Agile teams pull from the backlog, implement stories, integrate and test changes, and deliver a solution increment. These value-creation activities occur primarily during iteration execution. As an integral member of the team and their primary customer proxy, the PO provides daily insights that guide development toward the highest-value outputs and the team toward meeting iteration goals. This enables the team and, in turn, the ART, to deliver continuous value.

Balance stakeholder perspectives POs constantly receive input, feedback, and insights from customers, stakeholders, teams, and tools that can impact solution development. This information can validate, invalidate, or challenge implementation decisions unexpectedly. Moreover, these sources often conflict with one another. POs balance these perspectives by understanding the needs that drive them, remaining customer-centric, respecting capacity allocations, evaluating the cost of delay, and collaborating with stakeholders and teams to make implementation decisions that produce the most favorable outcomes.

Elaborate Stories Stories are typically created before iteration execution but require ongoing elaboration. POs facilitate frequent conversations with their teams to resolve questions, manage dependencies, and communicate priorities that emerge as stories are implemented. This information also helps the team slice stories effectively to achieve increased velocity and shortened learning cycles.

Foster Built-In Quality As the primary proxy for customers and stakeholders on the team, the PO plays a pivotal role in evaluating the value delivered from the backlog. The PO regularly evaluates progress toward story acceptance criteria, including compliance with [Built-In Quality](#) criteria, such as the scalable definition of done, and non-functional requirements (NFRs). The PO works closely with the team to detect quality issues as they are introduced and correct them in or near real-time.

Participate in team and ART events As a member of the Agile team, the PO, naturally, attends and actively participates in team events during PI execution. During iteration planning, backlog refinement, iteration reviews, team retrospectives, and team syncs, the PO provides crucial feedback on the team's work from an outside-

in, customer-centric point of view. By participating in [PO Sync](#) and [System Demos](#), the PO helps the team satisfy dependencies, demonstrate incremental value, and maintain cadence with the ART.

Getting and Applying Feedback

The PO is responsible for maximizing the value delivered by an Agile team. This, of course, implies that value is known. That knowledge comes from frequent feedback from customers and stakeholders—not just upon delivery but throughout the entire delivery life cycle. The PO is critical in enabling the continuous feedback loops that fuel the value stream. The PO seeks quantitative and qualitative feedback to develop a comprehensive understanding of where solutions are and are not providing real value. The following activities enable the PO to gather and apply feedback from several key sources:

Test benefit hypotheses ☰ Value can only be realized when a working solution is in the hands of the customer. Even then, it is not guaranteed. During development, therefore, value can only be speculated. To mitigate the risk of delivering solutions with little value, the PO collaborates with Product Management to define benefit hypotheses based on extensive customer and market knowledge. These hypotheses drive implementation and are validated (or invalidated) by feedback the PO gathers from customers and stakeholders throughout the product life cycle.

Obtain feedback from customers and stakeholders ☰ Customers derive value by using delivered solutions. Their feedback indicates how well solutions meet their needs, which drives solution adoption and loyalty. Stakeholders derive value from revenue, cost savings, or decreased risk stemming from customers’ use of delivered solutions. The PO gathers this feedback directly via empathy interviews, Gemba walks, iteration reviews, and system demos and indirectly via application telemetry, usage analytics, financial reporting, and secondary market data.

Share feedback with the ART ☰ Because solution delivery requires coordination and synchronization across the value stream, the feedback collected by the PO is valuable to the whole ART. The PO shares this information with Product Management and System Architects as part of [Continuous Exploration](#), with other POs during PO Sync, with their teams during backlog refinement, iteration planning, and iteration reviews, and with the ART during PI planning, system demos and, if applicable, [Inspect and Adapt](#) events.

Evolve solution design ☰ Frequent, customer-centric feedback cycles fuel the Plan-

Do-Check-Adjust cycle that enables continuous value delivery and the ongoing, relentless improvement of the value stream itself. By gathering and sharing these critical insights, the PO enables the continuous refinement of the product vision, roadmap, strategy, and design toward optimum business value.

Key Partnerships

The PO is ultimately responsible for maximizing the value delivered by the Agile team, which requires the PO to ensure that the *right solutions are built* and that they are *built the right way*. However, the PO cannot accomplish this alone.

Building the right solutions requires deep knowledge of business strategy, customer segmentation, market dynamics, and value stream economics. The PO establishes a close relationship with Product Management to derive these macro-level insights and apply them to specific product domains. Building solutions the right way requires [Team and Technical Agility](#), [DevOps](#) practices, and a [Continuous Delivery Pipeline](#). These technical capabilities determine the speed and quality with which value can be delivered, and the PO relies on the Agile team to provide them.

The PO provides a crucial link in the bi-directional information flow between Product Management and the Agile team. As shown in Figure 2, the PO keeps the Agile team informed of the strategy that drives product design and keeps Product Management informed of the innovations that influence the evolution of product strategy. Customer feedback aligns thinking from strategy through execution and is accessible to all roles.

Figure 2. Key PO relationships

Figure 2. Key PO relationships

Last update: 20 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



All it takes to play baseball is a strong arm, good speed, and coordination to hit the ball. That's it.

ÑRyne Sandberg

Value Stream Coordination

Value Stream Coordination describes how to manage dependencies between value streams and exploit the opportunities that exist in the interconnections.

Details

Value Streams are long-lived and generally independent of each other. For example, a systems or software company may sell many products and services, mainly decoupled from each other in technology. More likely, however, is that value streams have dependencies between them. Although we typically think of dependencies negatively, [Principle #2 – Systems Thinking](#) informs us that value flows through these dependencies. Yes, there are challenges to be addressed, but there are also valuable opportunities to exploit.

Most importantly, this additional value is often unique and differentiated, allowing an enterprise to offer solutions via dependencies that competitors cannot match. Or perhaps the competitor has not developed mastery in surfacing the emerging capabilities that these coordinated value streams can provide.

Exploiting opportunities from the interconnections between value streams requires the ability to coordinate value streams within a portfolio, as illustrated in Figure 1 and described in the sections below.

Figure 1. Cross-Value Stream coordination

Figure 1. Cross-Value Stream coordination

1. Coordination Roles

SAFe is anchored by people in three primary roles (or triads), representing a repeating pattern of the level below it.

What gets built: Product Owner > Product Management > Solution Management

How to build it: Agile Team > System Architect > Solution Architect

Servant leadership: Scrum Master/Team Coach > Release Train Engineer (RTE) > Solution Train Engineer (STE)

So, it isn't surprising to see similar roles and responsibilities appear in large portfolios whenever a significant degree of coordination is required, as shown in Figure 2 and described below:

Figure 2. Cross-value stream coordination roles

Figure 2. Cross-value stream coordination roles

Enterprise Architect provides technical guidance for the long-term evolution of the technologies and platforms and the more significant [Nonfunctional Requirements](#) (e.g., security, [Compliance](#), performance) for the portfolio's solutions.

Solution Portfolio Management is a function responsible for guiding a set of integrated solutions for a portfolio. One or more senior executives accountable for a solution set fulfills this function.

Value Management Office (VMO) and RTEs and STEs are typically responsible for supporting decentralized, efficient ART execution.

2. Apply Cadence and Synchronization

Figure 3 illustrates [Principle #7- Apply cadence and synchronization](#) at the [Portfolio](#) level to implement Business or Enabler Epics across development value streams.

Applying cadence and synchronization enables multiple value streams to collaborate to deliver portfolio-level initiatives (via [Epics](#)). This collaboration generally requires following a shared schedule for PI planning, integrated demos, and Inspect & Adapt events. Each demo enables the objective evaluation of the solution under development and supports large-scale continuous integration with internal and external suppliers.

Ideally, all value streams in a portfolio are aligned to a shared cadence, making routine things happen predictably. Since this cadence lowers the transaction costs associated with change, internal and external Suppliers should also be part of this

shared cadence.

Figure 3. Apply cadence and synchronization across dependent value streams

Figure 3. Apply cadence and synchronization across dependent value streams

3. Introducing New Portfolio Level Work

Figure 4 illustrates another vital concept. The portfolio cadence determines the rate and timing for adding new portfolio-level work. This cadence provides a reliable rhythm for introducing new portfolio work at PI boundaries since teams often cannot meet existing commitments and mix in significant unplanned work. It helps the [Agile Release Trains \(ARTs\)](#) achieve the enterprise predictability needs.

This cadence also establishes ways for [Epic Owners](#), [Enterprise Architects](#), and others to manage epics through the Portfolio Kanban system. Any epic not ready for [PI Planning](#) must wait for the following PI, even though capacity may have been available. Timeboxing the cadence also limits Work in Process (WIP) for the new and substantial work that will be introduced.

ARTs and [Solution Trains](#) focus on the committed PI Objectives during each PI. Any new work added to the system in the interim causes substantial interruptions, task switching, realignment, and movement of people to new objectives.

Figure 4. Introducing new portfolio-level work

Figure 4. Introducing new portfolio-level work

4. Ensured Integration Points

Integrating solutions across value streams is challenging. It may not be possible to do full integration in every iteration (Figure 5.) Therefore, it's imperative to do partial integration throughout the PI when full integration is not yet possible. These cadence-based integration points are the only objective measure of portfolio velocity—the more frequent the integration, the faster the learning.

Figure 5. Ensured integration points

Figure 5. Ensured integration points

It's essential to consider the following economic trade-offs when determining the cadence of integration and deployment:

- Benefits of faster learning, which often translates into higher quality and better products
- Cost of deferred learning and possible rework
- Cost and benefits of **DevOps** automation
- Depth of integration required
- Fidelity of feedback needed
- The level of customer satisfaction needed

Deployments are often subject to unpredictable surprises. Our natural inclination as humans is to delay tasks believed to be risky. Unfortunately, delaying deployments and creating large batches doesn't merely delay the risk. It increases the risk. If any given feature has a 1% chance of containing a defect that requires a production fix, then waiting to release a batch of 10 changes means you have nearly a 10% chance of needing a fix or rollback.

Moreover, letting small changes sit idle adds to the “holding” cost of the batch. Excess inventory is a form of waste. Instead, find a way to integrate these minor changes with others. A better choice is to lower the transaction cost of deployment by improving DevOps practices and, for example, adopting test and deployment automation, security and compliance automation, and using models to support end-to-end integration—demonstrating complete or partial system-level functionality.

5. Portfolio Roadmap

Figure 6 illustrates that the portfolio [Roadmap](#) is derived from and contributes to Solution roadmaps. This higher-level view allows the integration of relevant aspects of the solution roadmaps and their associated milestones into a more comprehensive view. This aggregated view communicates the larger picture to the enterprise and portfolio stakeholders.

Figure 6. Portfolio Roadmap

Figure 6. Portfolio Roadmap

6. Release on Demand

Deploying and releasing integrated value depends on effective [DevOps](#) and [Continuous Delivery Pipeline](#) capabilities due to the nature of value streams and dependencies. In some cases, ARTs provide all the DevOps capability that's needed. In others, there are additional considerations (Figure 7). These may even require dedicated or [Shared Services](#) and [System Teams](#) for individual ARTs and across value streams that help integrate the solution into a portfolio-level release.

Figure 7. Deploying and releasing work across value streams

Figure 7. Deploying and releasing work across value streams

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

Last update: 22 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*In the Age of
Digital, every
business is a
software business.*

*Agility isn't an
option, or a thing
just for technical
teams, it is a
business
imperative.*

—Dean Leffingwell,
Creator of SAFe

[Adjust Cookie Settings](#)

SAFe 6.0

SAFe is the world's leading framework for Business Agility. SAFe integrates the power of Lean, Agile, and DevOps into a comprehensive operating system that helps enterprises thrive in the digital age by delivering innovative products and services faster, more predictably, and with higher quality.

SAFe provides guidance on how businesses and employees can grow and respond to change. It started with the SAFe knowledge base and a training and certification program. Today, SAFe is in its sixth major iteration and has been adopted by more than 20,000 enterprises across the globe. More than 1,000,000 practitioners have been trained through a role-based curriculum and use the SAFe Community Platform, supported by more than 500 world-class transformation and platform companies.

Configurable and scalable, SAFe allows each organization to adapt the Framework to its business needs. With four out-of-the-box configurations, SAFe supports the full spectrum of solutions, from those requiring a small number of teams to those complex systems that require hundreds—and even thousands—of people to build and deliver.

Business Benefits of SAFe

Surviving in the age of digital is not guaranteed. Business agility isn't an option; it's imperative. Even businesses that don't consider themselves Information Technology (IT) or software companies—professional services, financial services, manufacturers, healthcare institutions, defense contractors, government agencies, and more—are now all highly dependent on their ability to rapidly produce new, high-quality, innovative, digitally-enabled products and services.

Improves Business Outcomes

Created from over a decade of field experience, SAFe draws from four primary bodies of knowledge: Agile development, systems thinking, Lean product development, and DevOps. It helps enterprises answer the following types of questions:

- How do we align technology development to strategic business goals?
- How do we deliver new value on a predictable schedule so the business can plan?
- How do we improve the quality of our solutions and delight our customers?
- How do we scale Agile practices from teams to ARTs across value streams and the

enterprise to deliver better results?

How do we organize people around value to deliver value effectively and avoid the delays inherent in a traditional, functional structure?

How do we create an environment that fosters collaboration, innovation, and relentless improvement for our people?

How can we change our culture so that it is safe to fail? How do we encourage people to take risks, think creatively, and embrace continuous learning? How can we help our teams improve without getting in the way?

By adopting SAFe[®] and applying its well-described set of values, principles, and practices[™] the enterprise can address these questions and realize more significant business and individual benefits.

SAFe 6 enables business agility and improves business outcomes for organizations of all sizes across the world. SAFe has produced dramatic improvements in time to market, employee engagement, higher quality, higher customer satisfaction, and improved economic outcomes. It also helps create more productive, rewarding, and fun cultures.

Figure 1 highlights these benefits derived directly from [Customer Stories](#) authored by SAFe customers.

Figure 1. SAFe business benefits as derived from SAFe customer stories

Figure 1. SAFe business benefits as derived from SAFe customer stories

SAFe Evolves Continuously

A hallmark of SAFe is that it stays current with new and evolving business and technology trends. Our incremental delivery model enables enterprises to adapt quickly and stay ahead of the competition by leveraging the latest knowledge in SAFe. For example, COVID-19 forced many organizations to change how Agile teams and ARTs collaborate. As a result, PI Planning, Organizational Agility, and Agile Teams[®] guidance were updated quickly.

Most updates to SAFe require changes to various articles and new advanced topics. However, Big Picture (BP) updates are sometimes needed to emphasize and provide ready access to the latest content. Minor releases of the BP occur under a [®]dot release like SAFe 5.1. Such minor releases are incremental and do not require upgrades to training and certification.

Over time, the incremental advances and significant new knowledge drive the

release of a new version, as was the case with SAFe 6.0. In these cases, SAFe practitioners and SPCs must keep pace with upgraded knowledge, training, and certification.

To help our community keep current with the latest guidance, we've introduced a new feature that provides a log of meaningful incremental changes on the home page under the heading, "What's New in SAFe." When you click on a change item, it will open a blog post and guidance article to inform our readers of the "why" and "what" was changed.

Introduction to SAFe 6.0 Big Picture

The SAFe website features an interactive Big Picture graphic. It provides a visual model of the Framework and is the primary user interface to the knowledgebase. Each icon of the image is clickable and offers access to extensive SAFe guidance. The configurations support a full range of development and business environments and the foundational principles, values, mindset, roles, artifacts, and implementation elements that make up the SAFe.

The main components of the SAFe are described in the following sections.

Figure 2. Full SAFe with configuration tabs

Overview Ð Seven Core Competencies

SAFe 6.0 features an Overview tab that illustrates the Seven Core Competencies (Figure 3) and their twenty-one dimensions that enable business agility. They are the *primary lens* for understanding and implementing SAFe. The focal point for all the competencies is the customer. Lean-Agile Leadership is the Foundation. This overview is valuable for providing an initial orientation to SAFe, introducing the business agility assessment, and framing executive briefings.

Figure 3. The SAFe Overview

The following competencies offer the knowledge, skills, and behaviors which enable enterprises to achieve business agility:

Team and Technical Agility

It all starts with Agile development, the cornerstone of Business Agility. The *Team and Technical Agility* competency describes the critical skills and Lean-Agile principles and practices that high-performing Agile teams and Teams of Agile teams use to create high-quality solutions for their customers. It consists of three

dimensions, as illustrated in Figure 4:

Figure 4. The three dimensions of Team and Technical Agility

Agile Teams ☐ High-performing, cross-functional teams anchor the competency by applying effective Agile principles and practices.

Built-in Quality ☐ All Agile teams share the goals and principles of [Built-in Quality](#). These Agile practices help create high-quality, well-designed solutions that support current and future business needs.

Team of Agile Teams ☐ Agile teams operate within the context of a SAFe Agile Release Train (ART), a long-lived team of Agile teams that provides a shared vision and direction and is ultimately responsible for delivering solutions.

Agile Product Delivery

Business Agility demands that enterprises rapidly increase their ability to deliver innovative products and services. To ensure that the enterprise creates the *right solutions* for the *right customers* at the *right time*, it must balance its *execution focus* with a *customer focus*. These mutually supportive capabilities make opportunities for sustained market and service leadership. **Agile Product Delivery** is a customer-centric approach to defining, building, and releasing a continuous flow of valuable products and services to customers and users.

There are three dimensions to Agile Product Delivery, as illustrated in Figure 5.

Figure 5: Three Dimensions of Agile Product Delivery

Customer Centricity and Design Thinking ☐ Puts the customer at the center of every decision. Applies design thinking to ensure the solution is desirable, feasible, viable, and sustainable.

Develop on Cadence, Release on Demand ☐ Developing on cadence helps manage the variability inherent in product development. Decoupling the release of value from that cadence ensures customers can get what they need when they need it.

DevOps and the Continuous Delivery Pipeline ☐ DevOps and the Continuous Delivery Pipeline create the foundation that enables enterprises to release value, in whole or in part, at any time it's needed.

Enterprise Solution Delivery

Building and evolving large enterprise solutions is a monumental effort. Many such systems require hundreds or thousands of engineers. They demand sophisticated, rigorous practices for engineering, operations, and support. Moreover, over the decades that these systems have been operational, their purpose and mission have evolved. That calls for new capabilities, technology upgrades, security patches, and other enhancements. As Òliving systems,Ó the activities above are never really Ôdone.Ó Instead, they are released earlier and further developed over time.

The *Enterprise Solution Delivery* competency describes applying Lean-Agile principles and practices to the specification, development, deployment, operation,

and evolution of the world's largest and most sophisticated software applications, networks, and cyber-physical systems. It consists of three dimensions. (Figure 6).

Figure 6. Three dimensions of Enterprise Solution Delivery

Lean Systems Engineering applies Lean-Agile practices to align and coordinate all the activities necessary to specify, architect, design, implement, test, deploy, evolve, and ultimately decommission these systems.

Coordinating Trains and Suppliers coordinates and aligns the extended set of value streams to a shared business and technology mission. It uses the coordinated Vision, Backlogs, and Roadmaps with shared PIs and synchronization points.

Continually Evolve Live Systems ensures that both the development pipeline and the large systems support continuous value delivery during and after release into the field.

Lean Portfolio Management

The three competencies above provide the technical practices needed to build and deploy meaningful business solutions. But none directly addresses the more significant issue of why those solutions are required, how they are funded and governed, and what other solutions are necessary to deliver total enterprise value. For that, we need to address portfolio concerns. However, traditional approaches to portfolio management were not designed for the impact of digital disruption. These factors pressure enterprises to work under a higher degree of uncertainty and deliver innovative solutions much faster. Portfolio Management approaches must be modernized to support the new Lean-Agile way of working. The *Lean Portfolio Management* competency aligns strategy and execution by applying Lean and systems thinking. As Figure 7 illustrates, it accomplishes this through three collaborations for strategy and investment funding, Agile portfolio operations, and

Lean Governance.

Figure 7. Lean Portfolio Management responsibilities

Strategy and Investment Funding ensures that the entire portfolio is aligned and funded to create and maintain the solutions needed to meet business targets. It requires the cooperation of Business Owners, portfolio stakeholders, technologists, and Enterprise Architects.

Agile Portfolio Operations coordinates and supports decentralized execution, enabling operational excellence. It requires the cooperation of the Value Management Office/Lean-Agile Center of Excellence (VMO/LACE) and Communities of Practice (CoPs) for Release Train Engineers (RTEs) and Scrum Masters/Team Coaches.

Lean Governance manages spending, audit and compliance, forecasting expenses, and measurement. It requires the engagement of the VMO/LACE, Business Owners, and Enterprise Architects.

Organizational Agility

Even with the competencies above, enterprises must be able to change quickly to respond to the challenges and opportunities that today's rapidly evolving markets present. This reality requires more flexibility and adaptability than the hierarchical operating system is likely to be able to muster. Again, we turn to the second

operating system for help. SAFe helps businesses address these challenges with *Organizational Agility*, which is expressed in three dimensions (Figure 8):

Figure 8. Three dimensions of Organizational Agility

Lean-Thinking People and Agile Teams Ø This state occurs when everyone involved in solution delivery is trained in Lean and Agile methods and embraces and embodies the values, principles, and practices.

Strategy Agility occurs when an organization demonstrates the ability and adaptability needed to sense the market and continuously change strategy when necessary.

Lean Business Operations Ø Teams apply Lean principles to understand, map, and continuously improve the business processes that support the business's products and services.

Continuous Learning Culture

And even with mastery of the above, there can be no final state. Startup companies will continue to challenge the status quo. Companies like Amazon and Google are entering new markets like banking and healthcare. Expectations from new generations of workers, customers, and society as a whole, challenge companies to think and act beyond balance sheets and quarterly earnings reports.

The *Continuous Learning Culture* competency describes a set of values and practices that encourage individuals and the enterprise to continually increase knowledge, competence, performance, and innovation. It is expressed in three dimensions, as shown in Figure 9.

Figure 9. The three dimensions of a continuous learning culture

The three dimensions are:

Learning Organization ☐ Employees at every level are learning and growing so that the organization can transform and adapt to an ever-changing world.

Relentless Improvement ☐ Every part of the enterprise focuses on continuously improving its solutions, products, and processes.

Innovation Culture ☐ Employees are encouraged and empowered to explore and implement creative ideas that enable future value delivery.

Lean-Agile Leadership

Finally, we recognize that an organization's managers, executives, and other leaders provide the foundation ultimately responsible for the adoption and success of Lean-Agile development and mastery of the competencies that lead to Business Agility. Only they have the authority to change and continuously improve the systems that govern how work is performed. Only they can create an environment that encourages high-performing Agile teams to flourish and produce value. Leaders, therefore, must internalize and model leaner ways of thinking and operating so that team members will learn from their example, coaching, and encouragement.

By helping leaders develop along three dimensions, as illustrated in Figure 10, organizations can establish the core competency of *Lean-Agile Leadership*.



Figure 10. Three dimensions of Lean-Agile Leadership

Mindset, Values, and Principles ☰ By embedding the Lean-Agile way of working in their beliefs, decisions, responses, and actions, leaders model the expected norm throughout the organization.

Leading by Example ☰ Leaders gain *earned authority* by modeling the desired behaviors for others to follow, inspiring them to incorporate the leader's example into their development journey.

Leading Change ☰ Leaders *lead* (rather than support) the transformation by creating the environment, preparing the people, and providing the necessary resources to realize the desired outcomes.

SAFe for Government

SAFe 6.0 continues to expand guidance for Lean-Agile adoption in [Government](#). Based on the Foundation and principles of SAFe, the guidance emphasizes:

- Building on a solid foundation of Lean-Agile mindset, values, principles, and practices
- Creating high-performing teams of teams of government and contractor personnel
- Aligning technology investments with agency strategy
- Transitioning from projects to a lean flow of epics
- Adopting Lean budgeting aligned to development value streams
- Applying Lean estimating and forecasting in cadence
- Modifying acquisition practices to enable Lean-Agile development and operations
- Building in quality and compliance
- Adapting governance practices to support agility and lean flow of value

The SAFe for Government article is a unique element within the Framework. It's a portal to several articles on SAFe adoption in the public sector, providing links to Agile government resources, videos, and events. Many are hard to find, and some are available only through our government portal. Our portal is a small way SAI gives back to the thousands of dedicated civil servants and their industry partners working tirelessly to bring agility to the Government.

SAFe Configurations

SAFe supports the full range of development environments with four out-of-the-box configurations. The configurations can be accessed using the tabs shown in Figure 1. Each is described in the following sections and contains a guidance article on flow.

Essential SAFe

Essential SAFe (Figure 11) is the basic building block for all SAFe configurations and is the most straightforward starting point for implementation. It builds on the principles and practices of [Lean-Agile Leadership](#), [Team and Technical Agility](#), [Agile Product Delivery](#), and [Continuous Learning Culture](#) competencies. SAFe is anchored by an organizational structure called the [Agile Release Train \(ART\)](#), where Agile teams and critical stakeholders are dedicated to a meaningful, ongoing solution mission.

[Essential SAFe](#) includes both the Agile team and ART constructs, as shown in Figure 11. The [Team Flow](#) article describes how Agile Teams deliver a continuous flow of value to the customer. The [ART Flow](#) article explains how a team of Agile Teams, working with their extended stakeholders, can get closer to their customers and build Continuous Delivery Pipelines(CDPs) to accelerate the delivery of valuable products and services.

Figure 11. Essential SAFe configuration

Figure 11. Essential SAFe configuration

Large Solution SAFe

The [Large Solution SAFe](#) configuration (Figure 12) introduces the [Enterprise Solution Delivery](#) competency, which supports those building the largest and most complex solutions that require multiple ARTs and Suppliers but do not require portfolio-level considerations. Such solution development is typical for aerospace and defense, automotive, and government industries, where the large solution—not portfolio governance—is the primary concern. The [Solution Train](#) organizational construct helps enterprises with the most significant challenges—building large-scale, multidisciplinary software, hardware, cyber-physical, and complex IT systems.

The [Solution Train Flow](#) article describes how multiple ARTs can work together with an aligned mission to build some of the world’s largest and most important systems under a lightweight and cooperative governance model. Developing these solutions requires additional roles, artifacts, events, and coordination, as Figure 12

illustrates and described in the [Large Solution](#) level article.

Figure 12. Large Solution SAFe configuration

Figure 12. Large Solution SAFe configuration

Portfolio SAFe

The [Portfolio SAFe](#) configuration is the minimum set of competencies and practices that can fully enable business agility. Portfolio SAFe provides two additional competencies, [Organizational Agility](#) and [Lean Portfolio Management beyond Essential SAFe](#).

The [LPM competency](#) aligns portfolio execution to enterprise strategy and organizes development around the flow of value through one or more value streams. The [Organizational Agility](#) competency extends Lean thinking and practice throughout the enterprise and enables strategy agility.

In addition to the competencies, Portfolio SAFe provides principles and practices

for portfolio strategy and investment funding, Agile portfolio operations, and Lean Governance. The [Portfolio Flow](#) article how to further accelerate the flow of the significant initiatives (epics) needed to accomplish the portfolio vision and advance the enterprise strategy. Figure 13 illustrates the additional portfolio-level roles, artifacts, and events. These elements are further described in the [Portfolio level](#) article.

Figure 13. Portfolio SAFe configuration

Figure 13. Portfolio SAFe configuration

Full SAFe

The Full SAFe configuration (Figure 14) is the most comprehensive configuration, including all seven core competencies for [Business Agility](#). The world's largest enterprises typically use it to maintain portfolios of large and complex solutions. In some cases, multiple instances of various SAFe configurations may be required.

Figure 14. Full SAFe configuration

Figure 14. Full SAFe configuration

Spanning Palette

The Spanning Palette contains various roles and artifacts that may apply to a specific Agile Team, ART, Large Solution, or Portfolio context. An essential element of SAFe's flexibility and adaptability, the spanning palette contains additional guidance elements for specific contexts. Figure 15 illustrates two versions of the spanning palette. The leftmost figure is used by Essential SAFe, while the rightmost one serves all other configurations. However, since SAFe is a Framework, enterprises can apply any item from the more extensive spanning palette to Essential SAFe.

Below is a brief description of each spanning palette element:

Vision The vision describes a future view of the solution to be developed, reflecting customer and

stakeholder needs and the Features and Capabilities proposed to address those needs.

OKRs When used within SAFe, OKRs can help to support the Core Values of transparency and alignment between the Enterprise and Portfolio strategy and the work of the Agile Release Trains and Agile Teams to deliver on this strategy. Additionally, OKRs can be applied to measure organizational improvement activities, including the desired outcomes for the SAFe transformation.

Roadmap The roadmap communicates planned ART and value stream deliverables and milestones over a timeline.

AI Artificial Intelligence (AI) describes a wide range of intelligent machines capable of performing tasks that typically require human intelligence. SAFe offers a foundational context for applying AI in the enterprise and explains how SAFe can accelerate the successful adoption of this advanced technology.

Shared Services This represents the specialty roles necessary for an ART or Solution Train's success but cannot be dedicated full-time to any specific train.

Figure 15. Spanning Palette

Community of Practice (CoP) A community of practice is an informal group of team members and other experts acting within the context of an ART or enterprise with a mission to share practical knowledge in one or more relevant domains.

System Team The system team is a unique Agile team that assists in building and using the continuous delivery pipeline and, where necessary, validating full end-to-end system performance.

Measure & Grow Measure and Grow is how portfolios evaluate their progress towards business agility and determine their next improvement steps. This evaluation is achieved through SAFe's three measurement domains, *Outcomes*, *Flow*, and *Competency*, providing a comprehensive yet simple model for measuring progress toward business agility.

Foundation

Anchored by the Lean-Agile leadership and continuous learning culture

competencies, the SAFe foundation contains the mindset, values, principles, and implementation guidance needed to implement SAFe practices and achieve business agility.

Each foundation element shown in Figure 16 is briefly described below.

Figure 16. SAFe Foundation

Figure 16. SAFe Foundation

Lean-Agile Mindset □ The Lean-Agile Mindset combines the beliefs, assumptions, attitudes, and actions of SAFe leaders and practitioners who embrace the concepts of the Agile Manifesto and Lean thinking. It's the personal, intellectual, and leadership foundation for adopting and applying SAFe principles and practices.

Core Values □ Four core values of Alignment, Transparency, Relentless Improvement, and Respect for People define the belief and value system for SAFe.

SAFe Principles □ SAFe practices are grounded in ten principles that synthesize Agile methods, Lean product development, DevOps, and systems thinking, coupled with decades of practical field experience.

Implementation Roadmap □ Implementing the changes necessary to become a

Lean-Agile technology enterprise is a substantial change for most companies. SAFe provides an implementation roadmap to help guide organizations on this journey.

SPC A SAFe Practice Consultant (SPC) is a change agent who combines their technical knowledge of SAFe with an intrinsic motivation to improve their company's software and systems development processes.

Last update: 13 March 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
 Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

Scaled Agile Framework

[SUBSCRIBE TO THE SAFe BLOG](#)

Say Hello to SAFe 6.0!

NEWS

March 15, 2023 [Leave A Comment](#)

By [Dean Leffingwell](#)

CATEGORIES

[NEWS \(3\)](#)

[SAFE FRAMEWORK UPDATES \(1\)](#)

CALENDAR

APRIL 2023

Our mission at Scaled Agile has always been to help those who build the world's most essential systems do so better, faster, at less cost—and with more enjoyment in the process. It's self-evident that a mission like that has no end. Especially now, as the unprecedented times of the pandemic, global economy, and supply chain issues have driven home a reality that has been decades in the making—enterprises are overwhelmed by constant change. And while...

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Ç Mar

Framework Promotions

RECENT POSTS

NEWS

SAFE FRAMEWORK UPDATES

March 6, 2023

[Leave A Comment](#)By [Dean Leffingwell](#)

Hi Folks, One of the coolest things about our mission at Scaled Agile is that the challenge of Ôhelping people build the worldÕs most important systemsÕ lays ahead of us and not behind. As we look into the future, we see our customer needs Ñand the complexity of the systems they buildÑgrowing, putting even greater demands on SAFe. To meet these demands, we are evolving the Framework team. My teammates are some of the smartest and.

[READ MORE](#)

SAY HELLO TO SAFE 6.0!

MARCH 15, 2023

FRAMEWORK PROMOTIONS

MARCH 6, 2023

EXCITING NEW BOOK: SAFE FOR DEVOPS PRACTITIONERS

MARCH 6, 2023

TRAINING

COURSE CALENDAR

ABOUT CERTIFICATION

BECOME A TRAINER

Exciting new book: SAFe for DevOps Practitioners

NEWS

March 6, 2023

[Leave A Comment](#)By [Harry Koehnemann](#)

If you attended a DevOps conference several years ago, you would have seen sessions filled with deep discussions on technical, architectural, and coding practices. Over time, the DevOps community has increased interest in other areas, including Lean-Agile methods, leadership, and organizational change. Indeed, for DevOps to succeed, the tooling and technical practices must integrate with the new ways of working. Organizations need to think of them holistically as we do in SAFe. In his new.

[READ MORE](#)

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

[Scaled Agile Framework](#)

SAFe Glossary

The SAFe glossary is a set of definitions for all SAFe Big Picture elements. The extended glossary provides definitions for additional terms used in the Framework. Some are unique to SAFe (e.g., PO Sync), while others are common in Lean-Agile development (e.g., MVP). They are provided here for clarity in their meaning in the context of SAFe. All extended glossary terms appear in the English configuration and will appear in other language configurations once translated.

[Select language](#)[Show Extended Terms](#)[Print Glossary](#)[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [R](#) | [S](#) | [T](#) | [V](#) | [W](#)[Adjust Cookie Settings](#)

Agile Product Delivery (APD)

The Agile Product Delivery (APD) competency is a customer-centric approach to defining, building, and releasing products and services in a continuous flow of value to customers and end-users.

Agile Release Train (ART)

The Agile Release Train (ART) is a long-lived team of Agile teams that incrementally develops, delivers, and often operates one or more solutions in a development value stream.

Agile Teams

An Agile Team is a cross-functional group of typically ten or fewer individuals with all the skills necessary to define, build, test, and deliver value to their customer.

AI (Artificial Intelligence)

Artificial Intelligence (AI) is a term used to describe a wide range of smart machines capable of performing tasks that typically required human intelligence. AI can be applied at all levels of SAFe to build intelligent customer solutions, automate value stream activities, and improve customer insights.

Architectural Runway

The Architectural Runway consists of the existing code, components, and technical infrastructure needed to implement near-term features with minimal redesign and delay.

ART Backlog

The ART Backlog is a Kanban system that is used to capture and manage the features and enablers intended to enhance the solution and extend its architectural runway.

ART Flow

ART Flow describes a state where an ART delivers a continuous flow of valuable features to the customer.

B

Big Data

Big Data refers to the roles and practices required to collect, manage, normalize

and deliver large datasets that help enterprises make more informed, fact-based decisions.

Built-In Quality

Built-In Quality is a set of practices to help ensure that the outputs of Agile teams in business and technology domains meet appropriate quality standards throughout the process of creating customer value.

Business Agility

Business Agility is the ability to compete and thrive in the digital age by quickly responding to market changes and emerging opportunities with innovative, digitally-enabled business solutions.

Business and Technology

Business and Technology describes the patterns that may be applied to realize Business Agility by applying SAFe principles and practices across the enterprise.

Business Owners

Business Owners (BOs) are key ART stakeholders who have the primary business and technical responsibility for return on investment (ROI), governance, and compliance.

C

CALMR

CALMR is a DevOps mindset that guides the ART toward achieving continuous value delivery by enhancing culture, automation, lean flow, measurement, and recovery.

Capabilities

A Capability represents large solution functionality whose implementation often spans multiple ARTs and is sized to be delivered within a PI.

Cloud

The Cloud represents virtual, on-demand processing and storage services used for cost-effective and scalable infrastructure and operations, implementation of the DevOps toolchain, and development and hosting of AI applications.

Communities of Practice (CoPs)

Communities of Practice (CoPs) are organized groups of people with a common interest in a specific technical or business domain. They regularly collaborate to share information, improve their skills, and actively work on advancing their knowledge of the domain.

Compliance

Compliance refers to the strategy, activities, and artifacts that allow teams to apply Lean-Agile development methods to build systems that have the highest possible quality, while simultaneously ensuring they meet regulatory, industry, and other relevant standards.

Continuous Delivery Pipeline (CDP)

The Continuous Delivery Pipeline (CDP) represents the workflows, activities, and automation needed to guide new functionality from ideation to an on-demand release of value.

Continuous Deployment (CD)

Continuous Deployment (CD) is an aspect of the Continuous Delivery Pipeline that automates the migration of new functionality from a staging environment to production, where it is made available for release.

Continuous Exploration (CE)

Continuous Exploration (CE) is an aspect of the Continuous Delivery Pipeline that drives innovation and fosters alignment on what should be built by continually exploring the market and customer needs, defining a vision, roadmap, and set of features for a solution.

Continuous Integration (CI)

Continuous Integration (CI) is an aspect of the Continuous Delivery Pipeline in which new functionality is developed, tested, integrated, and validated in preparation for deployment and release.

Continuous Learning Culture (CLC)

The Continuous Learning Culture (CLC) competency describes a set of values and practices that encourage individuals—and the enterprise as a whole—to continually increase knowledge, competence, performance, and innovation.

Coordinate and Deliver

Coordinate and Deliver describes the practices Solution Trains use to maintain the alignment and collaboration needed to continuously deliver value to large solution customers.

Core Values

The four Core Values of alignment, transparency, respect for people, and relentless improvement represent the foundational beliefs that are key to SAFe's effectiveness.

Customer

Customers are the ultimate beneficiaries of the value of the solutions created and maintained by a portfolio's value streams.

Customer Centricity

Customer Centricity is a mindset that focuses on creating positive experiences for the customer through the full set of products and services that the enterprise offers.

D

Design Thinking

Design Thinking is a customer-centric development process that creates desirable products that are profitable and sustainable over their lifecycle.

Development Value Streams

A Development Value Stream is the sequence of activities needed to convert a business hypothesis into a digitally-enabled solution that delivers customer value.

DevOps

DevOps is a mindset, culture, and set of technical practices that supports the integration, automation, and collaboration needed to effectively develop and operate a solution.

E

Enablers

Enablers are backlog items that extend the architectural runway of the solution under development or improve the performance of the development value stream.

Enterprise

The Enterprise represents the business entity to which each SAFe portfolio

belongs.

Enterprise Architect

The Enterprise Architect is responsible for establishing the portfolio's technology vision, strategy, and roadmap.

Enterprise Solution Delivery (ESD)

The Enterprise Solution Delivery (ESD) competency describes the practices necessary to apply SAFe principles and practices to the specification, development, operation, and evolution of the world's largest and most sophisticated software applications, networks, and cyber-physical systems.

Epic Owners

The Epic Owner is responsible for coordinating epics through the portfolio Kanban system.

Epics

An Epic is a significant solution development initiative.

Essential SAFe

Essential SAFe provides the minimal elements necessary for Agile Release Trains to deliver solutions and is the simplest starting point for implementation.

F

Features

A Feature represents solution functionality that delivers business value, fulfills a stakeholder need, and is sized to be delivered by an Agile Release Train within a PI.

I

Innovation and Planning Iteration

The Innovation and Planning (IP) Iteration is a unique, dedicated iteration that occurs every PI. It provides an estimating buffer for meeting PI Objectives and dedicated time for innovation, continuing education, PI Planning, and Inspect and Adapt (I&A) events.

Inspect and Adapt

The Inspect and Adapt (I&A) is a significant event held at the end of each PI, where the current state of the Solution is demonstrated and evaluated. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop.

Iteration

Iterations are a standard, fixed-duration timebox during which Agile Teams and ARTs individually and collectively deliver incremental customer value while working towards the PI objectives.

Iteration Planning

Iteration planning is a SAFe Scrum event where all team members determine how much of the Team Backlog they can commit to delivering during an upcoming Iteration. The team summarizes this work as a set of committed iteration goals.

Iteration Retrospective

The Iteration Retrospective is a regular event where the team members discuss the results of the iteration, review their practices, and identify ways to improve.

Iteration Review

The Iteration Review is a regular SAFe Scrum event where the team inspects the iteration increment, assesses progress, and adjusts the team backlog.

L

Large Solution SAFe

Large Solution SAFe is for enterprises building large and complex solutions that do not require portfolio concerns.

Lean Budget Guardrails

Lean Budget Guardrails describe the policies and practices for budgeting, spending, and governance for a specific portfolio.

Lean Budgets

Lean Budgets is a financial governance approach that funds value streams instead of projects, accelerating value delivery and reducing the overhead and costs associated with traditional project cost accounting.

Lean Portfolio Management (LPM)

The Lean Portfolio Management (LPM) competency aligns strategy and execution by applying Lean and systems thinking approaches to strategy and investment funding, Agile portfolio operations, and governance.

Lean User Experience (Lean UX)

Lean User Experience (Lean UX) is a team-based approach to building better products by focusing less on theoretically ideal design and more on iterative learning, overall user experience, and customer outcomes.

Lean-Agile Leadership (LAL)

The Lean-Agile Leadership (LAL) competency describes how leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential.

Lean-Agile Mindset

The Lean-Agile Mindset is the combination of beliefs, assumptions, attitudes, and actions of SAFe leaders and practitioners who embrace the concepts of Lean Thinking and the Agile Manifesto.

M

Measure and Grow

Measure and Grow is an approach SAFe enterprises use to evaluate progress towards Business Agility and determine improvement actions.

Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering (MBSE) is the practice of developing a set of related models that help define, design, simulate, and document a system under development.

N

Nonfunctional Requirements (NFRs)

Nonfunctional Requirements (NFRs) are system qualities that guide the design of the solution and often serve as constraints across the relevant backlogs.

O

Objectives and Key Results (OKRs)

Objectives and Key Results (OKRs) is a collaborative framework for establishing clear goals and measurable outcomes.

Operational Value Streams (OVS)

An Operational Value Stream (OVS) is the sequence of activities needed to deliver a product or service to a customer.

Organizational Agility (OA)

The Organizational Agility (OA) competency describes how Lean-thinking people and Agile teams across the enterprise optimize their business processes, evolve strategy with clear and decisive new commitments, and quickly adapt the organization as needed to capitalize on new opportunities.

P

Participatory Budgeting (PB)

Participatory Budgeting (PB) is a collaborative process for allocating the portfolio budget to its value streams.

PI Objectives

PI Objectives summarize the business and technical goals that teams and trains intend to achieve in the upcoming PI.

PI Planning

PI Planning is a cadence-based event for the entire ART that aligns teams and stakeholders to a shared mission and vision.

Planning Interval (PI)

A Planning Interval (PI) is a cadence-based timebox in which Agile Release Trains deliver continuous value to customers in alignment with PI Objectives.

Portfolio Backlog

The Portfolio Backlog is a Kanban system that is used to capture and manage the business and enabler epics intended to create and evolve the portfolio's

products, services, and solutions.

Portfolio Flow

Portfolio Flow describes a state where Lean Portfolio Management provides a continuous flow of new epics to Solution Trains and ARTs to achieve the portfolio's vision and strategic themes.

Portfolio SAFe

Portfolio SAFe provides strategy and investment funding, Agile portfolio operations, and Lean governance for one or more value streams.

Portfolio Vision

The Portfolio Vision describes the future state of a portfolio's value streams and solutions.

Pre-Plan

Pre-Plan describes the activities that align and prepare ARTs within a Solution Train for PI planning.

Product Management

Product Management is the function responsible for defining desirable, viable, feasible, and sustainable solutions that meet customer needs and supporting development across the product life cycle.

Product Owner (PO)

The Product Owner (PO) is the Agile team member primarily responsible for maximizing the value delivered by the team by ensuring that the team backlog is aligned with customer and stakeholder needs.

R

Release on Demand

Release on Demand is an aspect of the Continuous Delivery Pipeline that releases new functionality immediately or incrementally based on business and customer needs.

Release Train Engineer (RTE)

The Release Train Engineer (RTE) is a servant leader and ART coach who facilitates ART events and processes, and supports teams in delivering value.

Roadmap

The Roadmap is a schedule of events and milestones that forecasts and communicates planned solution deliverables over a time horizon.

S

SAFe

SAFe is the world's leading framework for Business Agility. SAFe integrates the power of Lean, Agile, and DevOps into a comprehensive operating system that helps enterprises thrive in the digital age by delivering innovative products and services faster, more predictably, and with higher quality.

SAFe for Government

SAFe for Government is a set of success patterns that help public sector organizations achieve better solution development outcomes by implementing SAFe Lean-Agile values, mindset, principles, and practices.

SAFe Implementation Roadmap

The SAFe Implementation Roadmap consists of an overview graphic and a 14-article series that describes a strategy and an ordered set of activities for successfully implementing SAFe.

SAFe Lean-Agile Principles

SAFe is based on ten immutable, underlying Lean-Agile principles. These tenets and economic concepts inspire and inform the roles and practices of SAFe.

SAFe Practice Consultants (SPCs)

SAFe Practice Consultants (SPCs) are certified change agents who combine their technical knowledge of SAFe with an intrinsic motivation to improve the company's software, systems, and Agile business processes.

SAFe Scrum

SAFe Scrum is an Agile method used by teams within an ART to deliver customer value in a short time box. SAFe Scrum teams use iterations, Kanban systems, and Scrum events to plan, execute, demonstrate, and retrospect their work.

SAFe Team Kanban

SAFe Team Kanban is an Agile method used by teams within an ART to continuously deliver value. SAFe Kanban teams apply a flow-based process to their

daily work and operate within the ART iteration cadence.

Scrum Master/Team Coach (SM/TC)

The SAFe Scrum Master/Team Coach (SM/TC) is a servant leader and coach for an Agile team who facilitates team events and processes, and supports teams and ARTs in delivering value.

Set-Based Design

Set-Based Design (SBD) is a Lean development practice that keeps requirements and design options flexible for as long as possible during the development process.

Shared Services

Shared Services represents the specialty roles, people, and services required for the success of an ART or Solution Train, but that are not dedicated full-time.

Solution

A Solution is a product, system, or service that provides value to internal or external customers.

Solution Architect

The Solution Architect is responsible for defining and communicating a shared technical and architectural vision for a Solution Train to help ensure the solution under development will be fit for its intended purpose.

Solution Context

Solution Context identifies the critical aspects of the environment in which a solution operates.

Solution Demo

The Solution Demo provides stakeholders an integrated view of the contributions of multiple ARTs and suppliers to obtain objective evidence of solution performance and to gather feedback.

Solution Intent

Solution Intent is the repository for storing, managing, and communicating the knowledge of current and intended solution behavior and design.

Solution Management

Solution Management is the function responsible for defining desirable, viable, feasible, and sustainable large solutions that meet customer needs and for supporting development across the solution life cycle.

Solution Train

The Solution Train is the organizational construct used to build large solutions that requires the coordination of multiple ARTs and suppliers.

Solution Train Backlog

The Solution Train Backlog is a Kanban system that is used to capture and manage the capabilities and enablers intended to enhance the large solution and extend its architectural runway.

Solution Train Engineer (STE)

The Solution Train Engineer (STE) is a servant leader and coach who facilitates Solution Train events and processes, coordinates the work of ARTs and Suppliers, and supports ARTs in delivering value.

Solution Train Flow

Solution Train Flow describes a state where a Solution Train delivers a continuous flow of valuable capabilities to the customer.

Stories

Stories are short descriptions of a small piece of desired functionality written from the user's perspective.

Strategic Themes

Strategic themes are portfolio-level business objectives that provide competitive differentiation and strategic advantage. They provide business context for portfolio strategy and decision-making, representing aspects of the enterprise's strategic intent.

Supplier

A Supplier is an internal or external organization that develops and delivers solution components, subsystems, or services to ARTs and development value streams.

System Architect

The System Architect is responsible for defining and communicating a shared technical and architectural vision for the solutions developed by an ART.

System Demo

The System Demo provides stakeholders an integrated view of new features for the most recent iteration delivered by all the teams on the ART. Each demo provides an objective measure of progress and the opportunity to give feedback.

System Team

The System Team is a specialized Agile team that assists in building and supporting the Agile development environment, including developing and maintaining the Continuous Delivery Pipeline. They may also support the integration of assets, end-to-end solution testing, DevOps mindset and practices, deployment, and release on demand.

T

Team and Technical Agility (TTA)

The Team and Technical Agility (TTA) competency describes the critical skills, principles, and practices that high-performing Agile teams on an Agile Release Train use to create high-quality solutions for their customers.

Team Backlog

The Team Backlog is a Kanban system that is used to capture and manage the user stories and enablers intended to enhance the solution.

Team Flow

Team Flow describes a state in which Agile teams deliver a continuous flow of value to the customer.

V

Value Stream Coordination

Value Stream Coordination describes how to manage dependencies between value streams and exploit the opportunities that exist in the interconnections.

Value Stream KPIs

Value Stream Key Performance Indicators (KPIs) are the quantifiable measures used to evaluate how a value stream performs against its business objectives.

Value Stream Management (VSM)

Value Stream Management (VSM) is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle.

Vision

The Vision is a description of the future state of the solution under development. It reflects customer and stakeholder needs and the features and capabilities proposed to meet those needs.

W

Weighted Shortest Job First (WSJF)

Weighted Shortest Job First (WSJF) is a prioritization model used to sequence work for maximum economic benefit. In SAFe, WSJF is estimated as the relative cost of delay divided by the relative job duration.

FRAMEWORK	PARTNER	CONTENT & TRADEMARKS	SCALED AGILE, INC
Download SAFe Posters & Graphics	Becoming a Partner Partner Directory	FAQs on how to use SAFe content and trademarks Permissions Form Usage and Permissions	CONTACT US 5400 Airport Blvd., Suite 300 Boulder, CO 80301 USA
Watch and download SAFe videos and presentations	GET SOCIAL Twitter Linkedin YouTube		BUSINESS HOURS Weekdays: 9am to 5pm Weekends: CLOSED
Blog			Privacy Policy Cookie Policy Your California Consumer Rights
TRAINING			
Course Calendar About Certification Become a Trainer			English

Scaled Agile Framework



Change is the only constant in life.

— Heraclitus, a Greek philosopher

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture. Read the SAFe 6.0 Upgrade FAQ [here](#) or the General SAFe 6.0 FAQ [here](#).

Adjust Cookie Settings

What's New in SAFe 6.0

We are delighted to announce that SAFe 6.0 offers the newest and most effective Lean-Agile practices, enabling forward-thinking enterprises to navigate today's rapidly changing challenges and opportunities.

Why

SAFe

6.0?

The

unprecedented times of the pandemic only emphasized a reality that has been decades in the making—Enterprises are overwhelmed by constant change. Between Customer expectations, supply chain complexities, and the impact of digital technology, the pace of change has never been faster, or the change itself has become more unpredictable.

For thousands of global organizations and government agencies, SAFe has become integral for adapting to change, building resiliency, and thriving as a modern, digital business. Meeting the new realities of the marketplace requires SAFe to evolve continuously. While we'll never lose sight of SAFe's core benefits, our guidance continually improves to address new and emerging challenges.

Work Differently. Build the Future.

Our new tagline, *“Work Differently. Build the Future.”* represents the idea that our work enables us to innovate faster to successfully navigate any market disruption or opportunity. What does it mean to work differently? For starters, it's increasingly clear—and now proven—that the entire organization needs to work with the same Lean-Agile mindset and practices. From marketing and finance to security and HR and everyone in between, this is how today's most successful enterprises align strategy to execution and meet current and future customer expectations.

Integrating emerging technologies (such as AI, Big Data, and Cloud) into our Value

Streams is a growing component of working differently. They are changing the nature of application development and will determine if an organization will succeed. Moreover, the pressure to accelerate the flow of value and deliver measurable business outcomes is unrelenting. The power of the Agile Team has never been more critical in achieving this.

Say Hello to SAFe 6.0!

SAFe 6.0 is a comprehensive update from version 5.1 and includes many new and advanced practices alongside a new Big Picture (BP) and terminology updates (Figure 1).

Figure 1. SAFe 6.0 Big Picture

The SAFe 6.0 update has six primary themes, as illustrated in Figure 2:

Figure 2. The six primary themes of the SAFe 6.0 update

Figure 2. The six primary themes of SAFe 6.0

Each of these themes is described in the following sections:

1. Strengthening the Foundation for Business Agility

Everything moves fast in the digital age. Customer desires, competitive threats, technology choices, business expectations, revenue opportunities, and workforce demands now happen at *blistering speeds*. To address these challenges, Business Agility is needed to compete and thrive in the digital age by quickly responding to market changes and emerging opportunities with innovative, digitally-enabled business solutions.

The fundamental aspects of business agility, including the **Business Agility Value Stream** (Figure 3) and **SAFe Foundation** (Figure 4), are substantially updated to

improve support for *working differently and building the future*. Updates to the SAFe Implementation Roadmap and revised responsibilities for SPCs clarify the path for navigating the change. Details include:

Business Agility Value Stream (BAVS) ▷ The newly introduced BAVS has been added to the top of the Big Picture, leading the way for competing and thriving in the digital age. It helps organizations quickly respond to market changes and emerging opportunities with innovative, digitally-enabled business solutions.

Figure 3. The Business Agility banner and BAVS

Figure 3. The Business agility banner and BAVS

SAFe Foundation Ñ All articles illustrated by the icons in Figure 4 were enhanced to further strengthen their support for business agility, along with the following design changes:

Figure 4. The SAFe foundation

Figure 4. The SAFe foundation

Lean-Agile Mindset ☐ The Lean-Agile mindset is represented by the five principles of *Lean Thinking* and the Agile Manifesto (Figure 5).

Figure 5. Lean-Agile Mindset

Figure 5. Lean-Agile Mindset

The House of Lean has been retired since all its tenets have been embodied and amplified throughout the Framework. For example:

Respect for people is now a SAFe Core Value since it's a basic human need. Treating people respectfully unlocks their intrinsic motivation to learn and grow, evolve their practices, and contribute to their business and customer outcomes.

Flow is covered in the new SAFe Principle #6 – Make value flow without interruptions and anchors Value Stream Management (VSM).

Innovation is part of the Continuous Learning Culture competency, which has been added to the SAFe foundation and will be expanded in future releases.

Relentless improvement became a SAFe core value since the relentless pursuit of perfection has always been one of the core tenets of Lean. While unattainable, striving for perfection leads to continuous improvements of products, services, and processes.

Core values – SAFe's four core values (Figure 6) were changed to

alignment, transparency, respect for people, and relentless improvement resulting from the previously described changes. Moreover, they were replaced to avoid redundancy with other concepts. For example, Built-in Quality is enhanced as a dimension of the Team and Technical Agility core competency, and an article with the same name.

Figure 6. Updated SAFe core values

SAFe Principles ☰ Principle #6 was changed and is now aligned with the five principles of Lean Thinking (Figure 7). It also highlights the eight common properties of a flow-based system and provides specific accelerators for eliminating impediments to flow.

Figure 7. SAFe Principles

SAFe Implementation Roadmap ☰ The SAFe Implementation Roadmap, which provides the critical moves for adopting the Framework, was updated to reflect the changes in SAFe 6.0 (Figure 8):

Figure 8. New implementation roadmap

1. **Waterfall / Ad Hoc Agile** step was removed as the starting point since SAFe adoption does not always begin from one of these starting points.
2. **SAFe Executive Workshop** toolkit was added to the "Go SAFe" step.
3. **Lean-Agile Center of Excellence** refined and clarified its responsibilities.
4. **Leading in the Digital Age** was added to the roadmap, a program that equips leaders with the knowledge and skills to support their Agile Teams and manage change effectively.
5. **Organize Around Value** was renamed from Identify ARTs and Value Streams to reflect its purpose and connection to Principle #10.
6. **SAFe Value Stream and ART identification** workshop toolkit was added to the "Organize Around Value" step.
7. **Enhance the Portfolio** was renamed. It was previously called Extend to the Portfolio. This updated step recommends that organizations start exploring some LPM practices earlier in their journey, such as implementing a Portfolio Kanban System to provide visibility of current and future initiatives. Therefore, LPM training is recommended during the "Train the executives, managers, and leaders" roadmap step.
8. **Accelerate** emphasizes that the transformation journey is never-ending, starting with a focus on creating a Continuous Learning Culture that commits to relentless improvement and promoting a culture of innovation.

SAFe Practice Consultants (SPCs) formerly known as SAFe Program Consultants, SPCs provide coaching for many of the *practices* and critical moves described in the SAFe Implementation Roadmap. They also must work beyond

the roadmap as a group of change agents with unrelenting energy, bringing business agility to the organization over time through executing the areas of responsibilities seen in Figure 9.

Figure 9. SAFe Practice Consultant (SPC) responsibilities

Figure 9. SAFe Practice Consultant (SPC) responsibilities

Continuous Learning Culture (CLC) ▷ The CLC competency was added to the foundation to make it part of all SAFe configurations. Previously, it was not part of the Essential SAFe configuration.

2. Empowering Teams and Clarifying Responsibilities

Nothing beats an Agile Team, and further enhancing performance requires clearer and more explicit responsibilities for each SAFe role. New and improved guidance is needed to help people improve their job performance and better support the organization's goals.

The following role articles were updated to illustrate and describe their responsibilities better. Selected thumbnails from these articles are highlighted in Figure 10.

- Agile Release Train
- Agile Teams
- Business Owners
- Enterprise Architect
- Epic Owners
- Lean Agile Center of Excellence
- Product Owner
- Product Management
- Release Train Engineer
- SAFe Practice Consultant
- Scrum Master/Team Coach
- Solution Management
- Solution Train Engineer
- System Architect

Figure 10. Selected thumbnails of the new responsibility areas

Scrum Master/Team Coach ☰ The Scrum Master/Team Coach now has broader responsibilities for optimizing flow, building high-performing teams, and supporting organizational agility (Figure 11). Organizations can choose the role name Scrum Master, or Team Coach, whichever is best suited for their context.

Figure 11. Scrum Master/Team Coach responsibilities

Agile Release Trains (ARTs) The ultimate purpose of ARTs is to deliver effective solutions to customers. It's fundamental for establishing a fast flow of solution features. ARTs develop solutions iteratively, constantly engaging with the customer and adjusting their course of action toward an optimal solution. Understanding their responsibilities is critical for success (Figure 12). To this end, a "responsibility wheel" was created for ARTs to clarify their duties at-a-glance.

Figure 13. Responsibilities of an ART

Figure 12. Responsibilities of an ART

Key collaborations ☰ Converting ideas into a steady stream of valuable products and services is a complex endeavor that requires ongoing coordination and communication. Four role articles with content authority were updated to include the collaborative partnerships needed to enable quick and efficient product development flow (Figure 13). Although these roles are integral to successful value delivery, no one has all the domain knowledge or capacity to do it alone.

Figure 13. Key collaborations for some roles with content authority

Figure 13. Key collaborations for some roles with content authority

3. Accelerating Value Flow

Being competitive in today's fast-paced world means accelerating value flow is an essential survival skill in the digital age.

SAFe 6.0 defines the eight properties of flow and, with an updated SAFe Principle #6, introduces eight related "flow accelerators" that can make value flow faster. Furthermore, brand new SAFe Scrum, SAFe Team Kanban, Built-in Quality, and Value Stream Management articles incorporate flow directly into the teams' daily work.

Eight common properties of flow describe how SAFe defines flow, as illustrated in Figure 14.

Figure 15. Flow systems have eight common properties

Figure 14. Flow systems have eight common properties

Eight flow accelerators  Figure 15 illustrates how the eight flow accelerators help make value flow without interruption (Principle #6). These powerful accelerators of value are relevant to all Framework levels, but the challenges differ for each.

Figure 15. The eight flow accelerators

Four new flow articles, Team Flow, ART Flow, Solution Train Flow, and Portfolio Flow, describe applying the eight flow accelerators from Principle #6. These articles can be accessed from the icons shown in Figure 16. Each offers techniques for addressing, optimizing, and debugging issues with a continuous value flow for that specific level.

Figure 16. Applying flow accelerators at each SAFe level

SAFe Team Kanban is a Lean-Agile method (Figure 17) that helps Agile Teams

facilitate the flow of value by visualizing workflow, establishing Work in Process (WIP) limits, delivering value continuously, measuring throughput, and improving their process. The article has been renamed from Kanban to ÔSAFe Team KanbanÕ and was significantly revised to explain how to use Kanban for SAFe teams, which operate as part of an ART.

Figure 18. SAFe Team Kanban method overview

Figure 17. SAFe Team Kanban method overview

SAFe Scrum replaces Scrum XP, and the XP-inspired practices were moved into Built-in Quality and other articles. In addition, the Scrum keystone image (Figure 18) was enhanced to illustrate how to apply Scrum in SAFe.

The 5.1 icons for Plan (Iteration Planning), Review (Iteration Review), Retro (Iteration Retrospective), and Execution were removed from BP and are now accessed from the Scrum article. The content from Iteration Execution was merged into the Iterations article.

Figure 19. SAFe Scrum Cycle

Figure 18. SAFe Scrum Cycle

Built-in Quality SAFe 6.0 provides a new innovative approach to built-in quality across all technology and business domains, illustrated in figure 19. This model starts with basic Agile Quality practices that apply universally. These are shift learning left, pairing and peer review, collective ownership, T-shaped skills, artifact standards and definition of done, and workflow automation. The model builds on this foundation with quality practices that apply to specific domains such as business functions, software applications, IT systems, hardware, and cyber-physical systems.

Figure 20. Built-In quality key domains and practices

Figure 19. Built-In Quality key domains and practices

Value Stream Management (VSM)  VSM was previously an extended guidance article. It was added to the Big Picture (Figure 20) to emphasize it as a Portfolio Level responsibility. It was also enhanced to provide new guidance about the triads that manage the value stream.

Figure 21. Value Stream Management on the Big Picture

Figure 20. Value Stream Management on the Big Picture

Value Management Office (VMO) ▷ The VMO was previously called the Agile Program Management Office (APMO), and its focus has changed to provide more support for understanding, measuring, and improving the flow of value.

The Value Management Office (VMO) supports portfolio flow, and its responsibilities include:

- Helps define the scope of the portfolio
- Supports improving the flow of value delivery
- Facilitates strategy formulation and helps communicate strategic themes
- Facilitates the Portfolio Sync
- Works with the LACE to develop, harvest, and apply successful ART execution patterns across the portfolio
- Facilitates Lean budgeting and governance
- Fosters decentralized PI Planning and operational excellence
- Fosters more Agile contracts and leaner supplier and Customer partnerships

Applying Kanban in SAFe ▷ Created a new Applying Kanban in SAFe extended

guidance article to provide an overview of Kanban, how to establish a Kanban system, and describe SAFe's connected Kanban system. Figure 21 offers some key images from this new content.

Figure 21. Applying Kanban in SAFe

4. Enhancing Business Agility with SAFe across the business

A Lean-Agile transformation will touch every part of the organization. As one part accelerates, it reveals bottlenecks in the others. Therefore, business agility requires everyone to learn how to perform their work better by applying Lean-Agile principles — optimizing flow, working with small batches, building-in quality, organizing around value, using systems thinking, and more — in their context.

Business & Technology — The significantly updated Business and Technology article highlights five business and technology patterns (Figure 22), including business-enabled ARTs, launching business trains, creating an Agile executive team, applying SAFe to other business functions, and combining development and operational value streams within the same portfolio. Each is described in the following paragraphs.

Figure 22. Extended business and technology patterns

Figure 22. Extended business and technology patterns

Business-Enabled ART □ An ART that includes the technical and business people needed to ensure the solution is aware of the industry in which it operates and that it addresses the relevant concerns for the technology, business, and customer.

Agile Business Train is a further step towards agility. An Agile Business Train contains one or more SAFe Operational Value Streams and all the ARTs needed to define, build, deploy, operate, and commercialize a complete business solution.

Agile Executive Team □ The Agile executive team aligns senior leadership across the organization to run the company as a team, communicate strategy and vision as one voice, and align on strategic Work-In-Progress.

Agile Functional Department is an organizational unit that utilizes Lean-Agile methods and SAFe principles to streamline its operations and create transparency in its value delivery.

Combined Portfolio □ A combined portfolio is a type of SAFe portfolio that includes both Development and Operational Value Streams. A combined portfolio provides clear strategic funding decisions and alignment for a blended

set of solutions within a business segment.

SAFe beyond IT home page provides access to articles about real-world business agility experiences in Operational Value Streams. Contributed and written by SAFe professionals, they describe their successes and challenges in extending Lean, Agile, and SAFe practices to Finance, Marketing, Human Resources, Legal, Operations, and other functions beyond IT.

5. Building the Future with AI, Big Data, and Cloud

SAFe offers new guidance on harnessing three innovative technologies that dramatically change the nature of software and systems development: AI, Big Data, and Cloud. Applying these technologies is becoming critical for competing now and in the future. There are three new articles and icons on the BP:

AI (Artificial intelligence) ▷ has been added to the spanning palette. It describes a wide range of intelligent machines capable of performing tasks that typically require human intelligence. It is a technology that can revolutionize solutions developed by SAFe organizations and has the potential to dramatically influence the operational and business models of enterprises as well.

Big Data ▷ SAFe addresses Big Data concerns at the portfolio level as it requires vision, investment, and governance at the highest levels in the organization. While ARTs create the data, the value comes from data aggregation at the portfolio and enterprise. Big data requires strategic investment from the most significant parts of the organization and a comprehensive approach that aligns each of the organization's development value streams to the same DataOps practices to produce cohesive data sets used across the entire organization.

Cloud ▷ The ever-expanding universe of cloud capabilities has fundamentally changed how digitally enabled solutions are built, deployed, and maintained. Cloud computing is the single most disruptive driver of delivery model change that enterprise IT has faced since its inception. Not surprisingly, the number one reason enterprises are moving to the cloud so quickly is to increase product development speed and agility.

6. Delivering Better Outcomes with Measure and Grow and OKRs

Measure and Grow ☰ The metrics icon on the spanning palette was renamed Measure & Grow. The content includes SAFe’s three measurement domains (*Outcomes, Flow, and Competency*), as illustrated in Figure 23. This approach enables measuring progress toward business agility at all levels of SAFe. The insights they provide support better decision-making and help identify opportunities for improvement.

Figure 24. SAFe’s three measurement domains

Figure 23. SAFe’s three measurement domains

OKRs were added to the spanning palette that links to a new guidance article. It provides three main use cases for OKRs illustrated in Figure 24 and described next. OKRs can offer an effective tool to help drive better outcomes for the business and customers. The usage of OKRs within SAFe is optional. However, OKRs are generally the best way to describe the portfolio’s Strategic Themes.

Figure 25. Use cases for applying OKRs in SAFe

Figure 24. Use cases for applying OKRs in SAFe

1. **Enhancing strategic alignment across a SAFe portfolio** Ⓛ The goal of using OKRs for Strategic Themes is to define and track their progress through concrete, specific, and *measurable* actions.
2. **Defining business outcomes for epics and Lean business cases** Ⓛ Use OKRs to help to uncover potential Epics for entry into the Portfolio Kanban system. Defining epics with OKRs definition provides clarity for the desired business outputs and outcomes.
3. **Setting improvement goals for the SAFe transformation** Ⓛ Applying OKRs can be used to measure the success of a SAFe change program.

Other important changes

A Kanban system manages all SAFe backlogs Ⓛ Each backlog is visualized in a Kanban system (Figure 25). Therefore, each backlog system and its related Kanban article have been merged to simplify guidance and remove redundancy. Other associated changes include:

Removing the *vertical Kanban label* from all backlog icons

Changing the Backlog state in all Kanban systems to "Ready," which reduces the confusion between the Kanban state and the backlog itself

Figure 25. Changes to the Kanban icons and backlog state

Spanning palette changes -The following changes were made to provide better support for SAFe 6.0 (Figure 26):

OKRs were added to the palette.

Milestones were merged into the new Roadmap article and removed from the palette.

AI (Artificial Intelligence) was added to the palette. It was previously an extended guidance article.

Lean UX was moved off the spanning palette to the body of the BP next to Customer Centricity and Design Thinking, showing its relationship and importance to those two practices.

Measure and Grow was renamed from metrics

Figure 26. Spanning palette changes

Enterprise Solution Delivery □ Based on field experience and customer feedback, the following articles were significantly improved to describe better how Solution Trains manage planning and deliver solutions (Figure 27):

Pre-Plan replaced Pre-PI planning. It describes the Solution Train activities necessary to prepare ARTs for PI Planning.

Coordinate and Deliver replaced Post-PI planning. This update describes the practices to deliver Large Solutions across multiple ARTs and suppliers in a Solution Train.

Figure 27. New Pre-Plan and Coordinate and Deliver practices for Enterprise Solution Delivery

Standardize ART terminology ☰ The terminology for program was replaced throughout the Framework with ART. Standardizing this terminology improves simplicity and clarity.

Table 1. Standardize ART terminology

Table 1. Standardized ART terminology

Standardize solution terminology □ Solution Train artifacts and events were standardized (Table 2).

Table 2. Standardized Solution Terminology

Table 2. Standardized Solution Terminology

Other terminology changes made for consistency are shown in Table 3 below.

Table 3. Other terminology changes

Last update: 2 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



When we start thinking about ways to line up all of the essential steps needed to get a job done into a steady, continuous flow, it changes everything.

Ñ James P. Womack and Daniel T. Jones, *Lean Thinking*

Accelerating Flow with SAFe

[Adjust Cookie Settings](#)

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Introduction

Everything moves fast in digital business. Customer desires. Business expectations. Revenue opportunities. Competitive threats. Technology choices. Workforce demands. And it's accelerating.

Today, markets move to where the value is—without hesitation, mercy, or guilt. To survive, enterprises must deliver continuous value in the form of products and experiences that engage and delight customers.

Major advances in technology are unlocking new ways to create this value. For example, AI, Big Data, Cloud, and DevOps enable enterprises to expand their product lines, modernize their existing offerings, scale to mass markets, make fact-based decisions, and streamline solution development. But to thrive in the digital age, building innovative solutions is not enough. It must be done fast, and that requires flow.

What is Flow?

Flow is characterized by a smooth transition of work through the entire value stream with a minimum of handoffs, delays, and rework. In SAFe, we consider flow to be present when teams, trains, and the portfolio can quickly, continuously, and efficiently deliver quality products and services from trigger to value (Figure 1).

Figure 1. Flow represents a smooth transition of work through the entire value stream

Flow requires all individuals and teams in the value stream to be tightly synchronized around value-creating activities. Otherwise, delivery is impeded by unnecessary interruptions. Indeed, flow is so critical that SAFe provides six specific flow metrics to measure how efficient an organization is at delivering value:

- Flow Distribution** ▷ The proportion of each backlog item type in the workflow
- Flow Velocity** ▷ The number of backlog items completed in a given time
- Flow Time** ▷ The time elapsed from when a backlog item enters the workflow to when it is released
- Flow Load** ▷ Total work in process (WIP) across all steps in the workflow
- Flow Efficiency** ▷ The amount of time backlog items are actively worked on as a percentage of the total time elapsed
- Flow Predictability** ▷ Overall planned versus actual business value

Understanding flow and the metrics used to measure them allows organizations to continually optimize their value streams for competitive advantage.

Flow Enables Business Agility

Enterprises must develop the ability to sense and respond to business opportunities and threats faster than the competition. SAFe enables that with the **Business Agility Value Stream** (BAVS) illustrated in Figure 2. This enables enterprises to not only deliver innovative solutions but to deliver them quickly and with maximum business value.

Figure 2. Flow enables the Business Agility Value Stream

Figure 2. Flow enables the Business Agility Value Stream

The key to achieving business agility is establishing a smooth and efficient flow of work through this entire process, from sensing an emerging opportunity to delivering the right solution. This requires all functions, processes, activities, teams, and events involved from end to end to be optimized for maximum speed and quality.

Implementing SAFe Establishes Flow

When implementing SAFe, organizations adopt a Lean-Agile mindset, identify value streams, organize around the flow of value, prioritize economically, develop on cadence, and release in a customer-centric manner.

These practices are inherently flow-based, so to implement SAFe is to establish a flow-based delivery system. For most organizations, this becomes a first major step change in delivery velocity over legacy, project-based methods.

Through this transition, organizations gain proficiency in SAFe's seven competencies, four of which are on the critical path for establishing flow:

Team and Technical Agility (TTA) ▷ The Lean-Agile principles and practices that

high-performing Agile teams and teams of Agile teams use to create high-quality solutions

Agile Product Delivery (APD) ▷ A customer-centric approach to defining, building, and releasing a continuous flow of valuable products and services

Enterprise Solution Delivery (ESD) ▷ Applying Lean-Agile principles and practices to the specification, development, deployment, operation, and evolution of the world's largest and most sophisticated software applications, networks, and cyber-physical systems

Lean Portfolio Management (LPM) ▷ Aligning strategy and execution by applying Lean and systems thinking approaches to strategy and investment funding, Agile portfolio operations, and governance

Developing these competencies removes the most wasteful activities from value streams, allowing them to begin to flow. SAFe's business agility assessment, individual competency assessments, and associated growth recommendations serve as a comprehensive guide to establishing this foundational level of efficiency.

Value Stream Management

Value Stream Management (VSM), is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle. VSM is the foundation of flow-based thinking in SAFe and is rooted in the following five Lean thinking principles:

1. Precisely specify value by specific product
2. Identify the value stream for each product
3. Make value flow without interruptions
4. Let the customer pull value from the producer
5. Pursue perfection

When applied across the portfolio, these principles foster a shared, Lean-Agile

mindset that focuses the organization on streamlining delivery. As described in the VSM article, each principle is supported by specific SAFe practices that enable smooth flow. Lean thinking principle #3 – Make value flow without interruptions, is the key to unlocking the potential for unprecedented speed.

Making Value Flow Without Interruptions

Step 3 in Lean thinking is the critical flow step. The end-to-end delivery system must be continuously cleared of wasteful activities that interrupt value flow. This requires organizations to develop the ability to systematically inspect value streams for inefficiencies, accurately diagnose root causes, and apply targeted improvements. SAFe Principle #6, [Make value flow without interruptions](#), provides a description of the mechanics of flow along with the following eight *flow accelerators*:

1. Visualize and limit WIP
2. Address bottlenecks
3. Minimize handoffs and dependencies
4. Get faster feedback
5. Work in smaller batches
6. Reduce queue length
7. Optimize time Ðin the zoneÑ
8. Remediate legacy policies and practices

Each flow accelerator is designed to address a specific type of delivery impediment. Because these impediments can manifest anywhere in the value stream, they apply at all levels of SAFe and are used by experienced Agilists to ÐdebugÑ flow issues and guide the portfolio toward continuous value delivery. Applying these accelerators thoughtfully and regularly is the key to achieving maximum delivery efficiency in SAFe.

Accelerating Flow for Teams, Trains, and the Portfolio

Because it is a flow-based system, SAFe instills a Lean-Agile mindset, Agile rhythm, and DevOps discipline throughout the organization. It unifies teams across functions and develops a culture of collective responsibility for achieving continuous value delivery. And when all teams and tools in the portfolio operate as a cohesive unit, unprecedented levels of acceleration can be achieved.

To realize this, flow interruptions are identified and removed wherever they occur.

Flow issues manifest differently in different contexts, so the eight flow accelerators must be regularly applied at all levels of SAFe to enable continuous delivery of value to the customer. Specifically, they are applied in unique ways at each level to enable:

Team Flow ▷ It all starts with a team's ability to deliver increments of value reliably, predictably, and continuously to its customers, stakeholders, and dependent teams. While Agile Teams are as cross-functional and autonomous as possible, they are not immune to interruptions to flow. Typical considerations include bottlenecks caused by insufficient T-skills, the tendency for individuals not to want to expose incomplete work to fast feedback, local optimization, and crushing loads of technical debt.

ART Flow ▷ Interruptions to flow can significantly hamper an Agile Release Train's ability to deliver integrated solutions with high speed and quality to its customers. Challenges faced by teams add up to produce larger bottlenecks at the ART level. For example, large batch sizes can delay customer feedback, clunky handoffs can delay integration, and legacy governance practices can delay deployment.

Solution Train Flow ▷ The coordination of multiple ARTs and Suppliers and the complexity of really big systems creates a ripe opportunity for new kinds of interruptions to occur. Systems are harder to fully test technically, and getting fast feedback from the end customer gets harder, large queues of committed future requirements can strip flexibility and responsiveness from the value stream, and legacy governance policies can slow down the system from the start.

Portfolio Flow ▷ A SAFe portfolio's ability to deliver a continuous flow of epics to achieve the portfolio vision, value stream KPIs, and business objectives can be heavily impacted by interruptions of many types. It can be hard for busy leaders to even find the time in zone to focus on this critical work. Early feedback can be hard to garner on intangible, emergent concepts, and lack of alignment can bottleneck critical decision-making.

Resolving flow issues at each level persistently and simultaneously ensures that there is nowhere for interruptions to hide. Flow is incrementally accelerated everywhere in the system, which continually speeds value delivery.

Of course, someone must be on hand to actively promote and manage ongoing flow acceleration. In SAFe, this important responsibility falls to coaches who develop and have skills to help diagnose and accelerate flow.

Coaching Flow

Enabling flow acceleration is a specialized skill that requires expert-level Lean-Agile practices applied throughout the organization. Leaders and practitioners are needed who can guide teams, ARTs, Solution Trains, and portfolios through the

process of continually optimizing the value stream.

Coaching flow involves these activities at all levels of SAFe:

Facilitating value stream mapping \triangleright [Value stream mapping](#) is fundamental to mastering flow. It enables the end-to-end delivery process to be visualized, and its efficiency measured, which creates a shared understanding of how work propagates through the organization and where flow needs to be improved.

Establishing Kanban systems \triangleright Every backlog in SAFe is supported by a [Kanban system](#), whose pull-based method of regulating flow promotes high throughput and quality.

Measuring flow \triangleright As described earlier, SAFe's [flow metrics](#) enable an organization to objectively monitor, evaluate, and improve the efficiency of its value streams.

Optimizing flow with the eight accelerators \triangleright The ongoing pursuit of perfection motivates an organization to [accelerate flow](#) and continually strive for a state of zero delay in the value stream. This requires constant attention to the natural impediments and interruptions to flow.

Thinking flow \triangleright All the work in applying flow practices and tools produces a sustainable, long-term improvement only if the new way of operating is anchored in the mindset and culture of the organization.

[SPCs](#) often lead the effort. As change agents and stewards of Lean-Agile practices, they move fluidly throughout all levels of SAFe to detect and resolve systemic flow issues. In addition, Scrum Masters, RTEs, STEs, the APMO, and the LACE all have a role in coaching the organization through the ongoing process of continuously and relentlessly accelerating flow.

Summary

Establishing and continuously accelerating flow is the key to achieving long-term, sustainable business agility. Implementing SAFe establishes a flow-based delivery system and culture. Value Stream Management builds upon that foundation by instilling Lean thinking across the organization. "Make value flow without interruptions" serves as a mantra and set of eight practices that enable flow issues to be detected and resolved at all levels of SAFe. Finally, ongoing coaching from experienced Lean-Agile leaders and practitioners fosters the most rapid acceleration.

Last update: 21 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

- Weekdays: 9am to 5pm
- Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



... select a winning contractor and then expect them to deliver on the requirements within the specified time frame and budget. However, this traditional approach almost always led to failures—each a spectacular waste of taxpayer dollars.

—Jason Bloomberg,
“Fixing Scheduling with Agile at the VA.” *Forbes*

[Adjust Cookie Settings](#)

Agile Contracts

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Builders of large-scale systems must continually align with [Customers](#) and other stakeholders on what's being built. And they often must do so amid continuous changes driven by development discoveries, evolving customer needs, changing technologies, and competitor innovations.

Traditionally, requirements and design decisions were made up-front to ensure Customers were getting what they wanted. That was the basis of the contract with the systems provider. But these early requirements and design decisions constrained teams, reducing their ability to adapt to emerging data that could have informed a [Solution](#) that could have delivered better economic and competitive value. In short, the contract held them back. Thus, attempts to manage risk by requiring early specificity often backfire to the disadvantage of all stakeholders.

Other contract approaches have evolved to avoid this, with more shared risk and reward. In many cases, they worked better. But even then, the conventional thinking of fixed requirements tends to influence agreements and expectations.

What's needed is a more Agile approach to contracts, one that benefits both parties in the near and long term. This article describes the current state and then guides an Agile Contract approach, the *SAFe Managed-Investment Contract*.

Details

Traditional Approaches to Purchasing Contracts for Systems

Buyers often outsource complex systems development to [Suppliers](#) who can build the systems needed to run their businesses. There's a range of approaches to contracting, from ***firm fixed prices*** to time and materials, with almost every point in between. Figure 1 characterizes these various approaches and highlights how the parties share risk.

Figure 1. A range of traditional contract types

Figure 1. A range of traditional contract types

However, almost everyone generally understands that neither extreme delivers the best overall economic value, as discussed in the sections below.

Firm Fixed-Price Contracts

On the left end of the scale are firm-fixed-price contracts, standard in today's industry. The convenience of this approach is the assumption that buyers will get exactly what they want and are willing to pay for, as Figure 2 illustrates.

Figure 2. Firm-fixed-price contracts create the ‘iron triangle’

Figure 2. Firm-fixed-price contracts create the Ôiron triangle.Õ

On the surface, this makes sense. In addition, it provides an opportunity for competitive bids, which may be required in many cases. In theory, competitive bids can offer economic advantages, as the request can go to the supplier with the lowest cost.

However, there are many downsides to this approach:

- It assumes that the buyerÕs needs are well understood before implementation.
- The buyerÕs needs must be reflected in early requirements specifications and design details. This triggers Big Design Up-front (BDUF) traditional development and contracts.
- The contract is typically awarded to the lowest-cost bidder, who may not provide the optimal long-term economic value for the buyer.

Moreover, critical decisions are made far too early to get a fixed bid when little knowledge about the solution is known (see [Principle #3 Ð Assume variability; preserve options](#)). The parties have entered into the Ôiron triangleÕ of fixed scope,

schedule, and cost, as illustrated in Figure 2. And if facts change, both the buyer's and supplier's hands are tied to the contract, which may now define something no one wants to build or buy precisely as was stated when written. Much of the time is spent negotiating contract changes, with significant waste.

Worst of all, once the agreement is entered, each party has an opposing economic interest:

It's in the buyer's short-term interest to get as much out of the supplier as possible for as little money as possible

Conversely, it's in the supplier's best short-term interest to deliver the minimum value necessary to meet contractual obligations and maximize supplier profits.

The net result is that this type of contract often sets up a win-lose scenario, which then influences the entire business relationship between the parties, typically to the detriment of both.

Time and Materials Contracts

It's clear why many would want to move to the right of the spectrum of approaches shown in Figure 1. But the time and materials agreements on the far right—which might appear to be highly Agile on the surface—also have their challenges. The buyer has only trust to count on. Trust is a precious commodity, and we depend on it in Lean. But misunderstandings, changes in the market or technical conditions, and changes in buyer or supplier economic models can force the trust to take a back seat. After all, it's in the supplier's financial interest to continue getting paid for as long as possible. This can drag contracts out for longer than necessary. Coupling this approach with a phase-gate process, whereby real progress can only be known at the end, compounds the problem.

Challenges can exist on the buyer's side as well. For example, when interviewed during a project postmortem, Stephen W. Warren, executive in charge and CIO of the Department of Veterans Affairs Office of Information and Technology, noted that according to the project manager, “the project was never in crisis since they were spending the entire budget every year, and thus were able to renew their funding for the following year. The measure of success at the time was whether the project would continue to get funding, rather than whether it could deliver the necessary functionality [1].”

A Collaborative Approach to Agile Contracts

Since neither endpoint in Figure 1 provides much assurance, perhaps the range in the middle is the sweet spot. Possibly, but even then, the biases of traditional

contracts—whether from the left or right in Figure 1—will likely creep into these agreements and expectations. What’s needed, then, is a different approach that trusts but verifies that the suppliers are building the right thing in the right way. Ideally, it provides regular and objective governance for the buyer yet allows suppliers to have confidence in their Customers and implied future economic commitments.

Characteristics of such an Agile contract would include the ability to:

- Optimize the economic value for all parties in the short- and long-term
- Exploit variability** via adaptive responses to requirements as new knowledge emerges
- Provide complete and continuous visibility and objective evidence of solution fitness
- Provide a measured approach to investment that can vary over time and stop when a sufficient value has been achieved
- Offer the supplier near-term confidence in funding and adequate notice when budgeting winds down or stops
- Motivate all parties to build the best solution possible within agreed-to economic boundaries

SAFe Managed-Investment Contracts

The industry would benefit by moving toward an Agile contracts approach, where the economics help both the buyer and the supplier. The “SAFe managed-investment contract” represents one such approach, as described below.

Pre-commitment

Before engaging in any significant investment contract for developing a complex system with many unknowns, some due diligence is required. In this case, the Customer and Supplier work together to come to terms based on the contract. This is the pre-commitment phase, illustrated in Figure 3.

Figure 3. SAFe managed-investment contract pre-commitment phase

Figure 3. SAFe managed-investment contract pre-commitment phase

During pre-commitment, the Customer has specific responsibilities, including understanding this Agile contract's basic constructs and obligations and defining and communicating the more comprehensive program mission statement to the potential supplier(s).

The supplier does their initial homework as well. This often includes the first analysis of potential feasibility and alignment of the buyer's solution needs with the supplier's core competence. It also demands an understanding of the personnel and resources required over the initial contract periods and a rough cost estimate.

The shared responsibilities, illustrated in Figure 3, start the Customer and supplier toward a more measured investment, supported by continuous objective evidence of fitness for use. These responsibilities include:

Establishing the initial Vision and Roadmap

Identifying the Minimum Viable Product (MVP) and additional features

Defining the initial fixed and variable Solution Intent
Prioritizing the initial ART Backlog for PI Planning
Establishing execution responsibilities
Establishing the [Economic Framework](#), including economic trade-off parameters, the PI funding commitment (number of PIs committed), initial funding levels, and other contractual terms

Sometimes, the supplier may need to provide a preliminary estimate to secure the PI funding commitment for completion. In other cases, a pay-as-you-go approach may be suitable. Based on the terms, the Customer will agree to fund the supplier for the early PIs. This is the initial commitment period. The length depends on context, but two PIs (~20 weeks) may be a reasonable starting point.

Depending on the context, the Customer may discuss with several potential suppliers. If significant technical feasibility is involved, this can often be done under some form of feasibility contract, whereby each potential supplier is compensated for the efforts to get to commitment. This may be business as usual for the supplier, with these pre-commitment investments regularly occurring in presale activities.

However, the Customer can move on to award the contract at some point.

Contract Execution

After the contract is awarded, development begins, as illustrated in Figure 4.

Figure 4. SAFe managed-investment contract execution phase

Figure 4. SAFe managed-investment contract execution phase

A description of the activity timeline follows:

PI preparation □ Both supplier and Customer will invest time and effort in preparing content and logistics for the first PI planning session. (Note: In some cases, it might be suitable that the first PI planning is part of the pre-commitment phase, though this route requires significant investment by both parties.)

PI planning □ The first PI planning event influences the entire program. There, customer and supplier stakeholders plan the first PI in iteration-level detail.

PI execution □ Depending on the context, Customers participate at various levels in iteration execution. At a minimum, however, each **System Demo** requires direct Customer engagement. For large solutions, however, system demos may be replaced by a more fully integrated Solution Demo, which can occur more frequently than at PI boundaries.

PI evaluation □ Each PI marks a critical Milestone for the Customer and the supplier. At each milestone, the solution demo is held, and the solution is evaluated. Agreed-to metrics are compiled and analyzed, and decisions are made for the following PI. The Inspect and Adapt (I&A) event assesses solution progress and

program improvements. At this point, the Customer may decide to increase, decrease, or maintain funding levels or wind down the initiative based on whether a sufficient value has been achieved. After that, the next PI planning commences, with the scope based on the outcome of that decision.

Managing Risk with the Lean Startup Approach

The SAFe Lean Startup Cycle shown in Figure 5 is also used to illustrate how significant product development investments are managed with sound, Lean economics.

Figure 5. The Lean Startup Cycle for managing significant product development investments

This Lean Startup model decreases time-to-market and helps prevent the system from becoming bloated with unnecessary features that may never be used. It also enforces the Ôhypothesize-build-measure-learnÕ cycle described in the [Epic](#) and [Continuous Delivery Pipeline](#) articles.

The implication is that Agile contract language is modified to reflect a combination of fixed and variable components. The MVP identified at the pre-commitment stage can establish a high-level definition of fixed scope to be delivered over a proposed number of PIs. Beyond the delivery of the MVP, the contract can also specify the number of option periods consisting of one or more PIs. The goal is to optimize the delivery of prioritized features within each PI.

This process continues until the solution has delivered the value the Customer requires. At this point, the Customer stops exercising additional option periods and starts winding down funding commitments by the agreement. This provides the best of both worlds to Customers:

Better predictability of estimates associated with a smaller MVP than the complete list of all requirements

Total control over the spending required for additional incremental features based on economic outcomes

Such an approach provides the most significant economic benefit to both parties, which will help create stable, long-term relationships.

Learn More

[1] Bloomberg, Jason. *Fixing scheduling with Agile at the VA*. Forbes, October 23, 2014.

[2] Jemilo, Drew. *Agile Contracts: Blast Off to a Zone of Collaborative Systems Building*. Agile 2015. <https://www.slideshare.net/JEMILOD/agile-contracts-by-drew-jemilo-agile2015>

Last update: 22 February 2023

FRAMEWORK

[Download SAFe Posters & Graphics](#)

Watch and download SAFe videos and presentations

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework

SAFe for Marketing

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Scaled Agile Inc. (SAI) has extended [SAFe beyond IT](#) and from the realm of pure tech teams to other teams in the business that are critical to the building and supporting of innovative business solutions. Moreover, the [Business and Technology](#) article describes the patterns that may be applied to realize [Business Agility](#) by applying SAFe principles and practices across the enterprise.

[Adjust Cookie Settings](#)

Integral to that, SAI has been working with a number of enterprises to integrate

marketing with their SAFe implementations. As we run our entire business on SAFe, we have our own business experience to share.

To this end, the SAFe for Marketing whitepaper (download below) provides insights on:

- What is driving the adoption of agility in marketing organizations
- The three ways marketing teams can integrate with SAFe: being Agile, joining the value stream, and specializing in principles and practices
- Using validated learning and hypothesis-driven approaches to optimize team and campaign performance

This white paper was authored by our own Melissa Reeve, Scaled Agile's former VP of Marketing. Melissa notes:

Since I arrived at Scaled Agile, I've been thinking about how to help others use the Scaled Agile Framework in marketing environments.

Enterprises around the world struggle to keep up with the relentless pace of change. Software and product development organizations have adopted Agile frameworks such as SAFe to help them address these challenges and improve time-to-market, productivity, and quality.

Marketing organizations are no exception. They face issues around alignment, inefficiencies, and slow reaction times. The typical product marketing campaign takes months to get to market, contributing to an estimated \$958 million in waste in B2B marketing organizations on an annual basis. To fully realize the benefits of business agility, marketing organizations must also adopt Agile.

We wrote this white paper for marketing teams and team leaders, for technical teams working closely with marketing teams, and anyone wanting to know how to better integrate marketing in their SAFe implementation.

We hope you find this a valuable resource that will get you one step closer to true business agility.

Download

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Kanban, you build a map of your work. The landscape depicted is your value stream. A value stream visually represents the flow of your work from its beginning through to its completion.

—Jim Benson, Personal Kanban: Mapping Work Navigating Life

[Adjust Cookie Settings](#)

Applying Kanban in SAFe

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Kanban is a Lean workflow management method that Agile teams use to define, manage, and continuously improve the products and services they build for customers. This method helps teams visualize and understand their workflow, optimize efficiency, and improve relentlessly.

Ø Kanban is a flow-based system. As most workflows exist to optimize value, the strategy of Kanban is to optimize value by optimizing flow. Optimization does not necessarily imply maximization. Rather, value optimization means finding the right balance of effectiveness, efficiency, and predictability in how work gets done [1].

Kanban's unprecedented level of visibility is causing it to spread to different parts of the organization. Today, many organizations adopt Kanban to help embrace Lean-Agile principles across all aspects of business, from marketing to finance, human resources to legal, security to compliance, operations to [Agile Teams](#), and more.

Details

In SAFe, Kanban systems manage the backlog and flow of work at every level of the Framework. Each reflects a team's unique process for delivering value and its current workflow and capacity.

Kanban System

A Kanban system is generally characterized by a "Kanban board" that includes or references the elements shown in Figure 1.

Figure 1. Elements of an example Kanban system

Figure 1. Elements of an example Kanban system

Work in Process (WIP) limits set the maximum number of items in an individual workflow state.

Columns represent a series of steps (states) representing an activity that collectively defines the team's workflow. (The steps above are just illustrative).

Cards represent the individual work items, such as stories, features, capabilities, and Epics, including enablers.

Swim lanes group and highlight related work items to further define the team's workflow. Typical use of swim lanes includes separating work for different classes of service and individual responsibilities, cross-team dependencies, and more.

Policies specify how work is managed, such as exit or entry criteria for moving a work item from one state to another or defining the rules for service classes.

Establishing a Kanban System

Implementing an effective Kanban system adapted to meet the needs of a specific Agile Team is based on the type of work performed (for example, software development, hardware design, marketing), the team members' skills, and the team's role in the enterprise.

Establishing a Kanban system is best done by involving the entire team with the guidance and facilitation of an experienced coach. The following tenets will help teams adopt Kanban [1]:

1. Start with what you do now
2. Agree to pursue incremental, evolutionary change
3. Respect the current process, roles, responsibilities, and titles
4. Encourage acts of leadership at all levels in your organization

The first step is critical for establishing the Kanban system and deserves additional clarification. The Kanban Method does not require changing the existing workflow. This approach makes implementation easier since the current processes do not have to be changed immediately. Kanban advocates gradual, evolutionary change based on empiricism. The team incrementally makes process improvements. With these tenets in mind, the initial design of the Kanban typically involves the six activities defined below.

1. Map the Team's workflow

A facilitator guides the team to map their current workflow to get started. It's important to note where work is handed off, as they may be candidates for creating buffer states to smooth the workflow. Figure 2 depicts a typical flow for a software team. Technical and business teams may follow a similar process. However, their process steps must be adapted based on the type of work performed.

Figure 2. An example of a software team's current workflow

Figure 2. An example of a software team's current workflow

2. Arrange the Workflow Steps

The current workflow informs the steps the team wants to track in the team Kanban. Once the team agrees on the existing workflow, they can arrange these steps on a Kanban board, but they may not be identical to the team's current flow. For example, some steps may be merged or split, and buffers or review states can be added. After reviewing the initial Kanban board, the team may decide to simplify it or add more steps. A Kanban with too many steps can make it overly complex, while too few can hide bottlenecks and non-value-added steps.

3. Identify Buffer states

Introduce buffer states to help manage variability in the team's workflow. Buffers show bottlenecks and delays in the system. Since each extra item of WIP carries a penalty in terms of lead time, start with a small limit and adjust it up or down based on observation. Reducing the variability in the size of the work items may allow you to reduce the buffer size.

4. Create Policies

A Kanban board makes the team's processes and policies explicit. For example, each state's entry or exit policies clarify what the team needs to do before a story can be pulled into the next state.

Below are some examples of explicit policies:

- The definition of done (DoD) for a work item
- Who can add, change, and prioritize the backlog?
- How to handle emergency requests
- What to do when team members become blocked from doing their work
- Who can move an item to the next state?

This list is not exhaustive. However, it provides example policies to help teams form their own. Have regular meetings with the team after specific milestones and revisit them so they can evolve. By viewing the Kanban board, the team and their stakeholders can have a shared understanding of their workflow, policies, and the status of its WIP.

5. Assign Initial WIP limits

At this point, the basic structure of the board is ready, and it is time to define the initial WIP limits. These limits are based on the team's experience with their current process, and this is often the first attempt to limit WIP at each step to facilitate a faster flow. Some states (like "Funnel" and "Done") usually do not need WIP limits.

Figure 3 shows an example of a team's workflow expressed as a Kanban board after mapping their workflow, arranging the steps, refining their workflow with buffers, and establishing WIP limits.

Figure 3. An example of one team's Kanban board

Figure 3. An example of one team's Kanban board

6. Identify Classes of Service

Kanban classes of service have two primary purposes: categorizing work items according to their priority and the ability to specify different individual policies for a specific work item type. The team agrees to establish and follow a particular execution policy for each class to optimize flow and value. Often teams start with a simple group of service classes, as shown in Figure 4:

Figure 4. Classes of service on the Kanban board

Figure 4. Classes of service on the Kanban board

Standard. Represents the baseline class of service, applicable to work items that are neither expedited nor have a fixed date and should not violate WIP limits.

Expedite. This class is for unexpected but urgent work with a high cost of delay and therefore requires immediate attention. As a result, items in this swimlane can be pulled into development, even if it violates current WIP limits. Therefore, teams should have policies limiting this class of service (for example, only one expedited item can be in the system at a time). Moreover, teams may set a policy to ÔswarmÖ expedited items to make sure it moves through the system rapidly.

Fixed date. Describes work items that must be delivered on or before a specific date. These items must be specifically identified and actively managed to mitigate schedule risk. Among other uses, this service class can ensure that flow-based Kanban teams meet dependency commitments.

A Kanban board should evolve iteratively and continuously adapt to fit the teamÔs needs. After defining the initial process and WIP limits and executing for a while, bottlenecks should become visible. If not, the team refines the process or further reduces some WIP limits until it becomes evident that a workflow state is

overloaded or starving. Other changes to optimize flow might include merging or splitting steps, adding buffers, or redefining workflow states.

Connected Kanban Systems in SAFe

Kanban systems are used throughout SAFe, including the Portfolio, Large Solution, and Essential levels (ART and Team), as illustrated in Figure 5. Each Kanban system has several things in common to help improve the flow of value. For example, they:

- help match demand to capacity based on Work in Process (WIP) limits
- help identify opportunities for relentless improvement by visualizing bottlenecks in each process state
- facilitate flow with policies governing the entry and exit of work items in each state

Figure 5. Connected Kanban systems in SAFe

Figure 5 illustrates how new work goes through the various Kanban systems and backlogs that manage the workflow. Kanban systems help strategy changes move quickly across value streams to the implementation teams. This way, execution is aligned—and constantly realigned—to the evolving business strategy.

However, not all work originates from the portfolio. Smaller, local changes are often needed, which may only require new stories, features, or capabilities. These local concerns will go directly into the appropriate Team, ART, or Solution Train backlogs.

The following sections briefly describe SAFe's four Kanban systems. A more detailed article supports each Kanban system.

Team Kanban

The [Team Backlog Kanban](#) system contains user and enabler stories from the ART Backlog and stories that arise locally from the team's context. It may also include other work items, representing everything a team needs to do to advance their portion of the system. The [Product Owner \(PO\)](#) or another similar role manages and prioritizes the Team Kanban and backlog.

ART and Solution Train Kanban Systems

The ART and Solution Train Kanban systems (described in the [ART and Solution Train Backlog](#) article) facilitate the flow of business and enabler [Features and Capabilities](#) through the [Continuous Delivery Pipeline](#).

[Product Management](#) and [Solution Management](#) prioritize and manage the ART and Solution Train Kanban systems. Features lend themselves to the [Lean UX](#) process model, including defining the Minimum Marketable Feature (MMF), a benefit hypothesis, and acceptance criteria. The MMF helps limit scope and investment, enhances agility, and provides fast feedback. Capabilities behave the same way as features. However, they are at a higher level of abstraction and support the definition and development of large [Solutions](#).

Portfolio Kanban System

The [Portfolio Kanban](#) is critical because it helps align strategy and execution by identifying, communicating, and governing the selection of the largest and most strategic initiatives (epics) for a SAFe portfolio. [Lean Portfolio Management \(LPM\)](#) operates the portfolio Kanban system, which uses the *strategic portfolio review* and *portfolio sync* events to prioritize, manage and monitor strategic workflow.

Learn More

[1] <https://kanbanguides.org/english/>

[2] Anderson, David J. *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, Washington: Blue Hole Press, 2010.

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



When we own the horizontal and vertical, we get to move, make changes, and respond to customer requirements at the pace we're used to. We think this is a really big deal.

~James Hamilton, an AWS VP and Distinguished Engineer

Applying SAFe to Hardware

Adjust Cookie Settings

Development

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

[Business Agility](#) requires everyone involved in product delivery to use Lean and Agile practices to create innovative, high-quality products. As Agile adoption extends beyond software, hardware development needs to keep pace. Although the hardware community is relatively new to Agile, SAFe's [Lean-Agile Values](#) and [Principles](#) are universal. They can be used to guide hardware engineers to create and adopt their own best practices.

An Agile transformation will ultimately affect every part of the [Enterprise](#). Twenty years ago, businesses struggled to deliver value to their customers due to bottlenecks in software development. Software practitioners began applying Agile practices and created new technologies such as virtualization, microservices, and infrastructure-as-code, accelerating execution and driving innovations. Today, organizations that employ these Agile practices and development innovations deliver value significantly faster with higher quality. And those early innovators now dominate the software marketplace.

Organizations building hardware systems now find themselves in a similar position. However, they can now apply the insights from the previous two decades. Many companies have already started this journey (Figure 1). Through extensive use of virtualization and learning in the digital world, General Motors cut the Hummer EV's launch time in half [1]. This is because digital engineering shifts learning to the left by integrating virtual drawings, models, and simulations to get feedback before creating physical parts.

SpaceX uses additive manufacturing on all parts of its system, from rocket engines to helmets [2]. Additive manufacturing prints physical parts on-premises directly from computer-aided design (CAD) data, on-premises, faster than traditional manufacturing and assembly processes. Apple, Google, and Amazon also design their hardware to meet business demands [3].

Figure 1. Hardware innovators are following the same path as software innovators

Details

To create innovative, high-quality products—including those with hardware elements—business agility requires hardware participation in the SAFe transformation. The following sections describe how SAFe applies to hardware development, outlining six universal principles and how to use them.

Organize Around Value

SAFe Principle #10 – Organize Around Value, states that an organization’s value flows across its functional silos. Traditionally, work is organized around functional skills. Take, for example, a new rocket nozzle that has a mechanical structure, which is hardware (HW)—with a custom electrical circuit design, which is the firmware (FW)—and control logic to adjust the nozzle, which is software (SW). With a traditional approach, each would be developed independently, aligned by detailed specifications and a standard schedule. Verification and validation can only occur when all the components are integrated at the end, making any adjustments costly.

To enable collaboration and reduce handoffs and delays, Agile development organizes teams of cross-functional skills. Figure 2 shows three equally viable ways where this might occur and the situations in which they are optimal:

Figure 2. Three examples of organizing teams with hardware development

Figure 2. Three examples of organizing teams with hardware development

1. Cross-functional Agile teams work in a highly innovative environment, with frequent experimentation and tight collaboration across all domains. Innovations in noise, electrical, and control logic design are quickly integrated and validated.
2. Some innovative environments do not require the tight collaboration of a single team, and the domains can work more independently. However, to ensure their work is aligned, both teams are part of the same Agile Release Train (ART) and use SAFe practices to manage dependencies and integrate frequently.
3. Other environments are more predictable. These teams can work independently on their system components, aligning their efforts with a joint [Roadmap](#) defining the critical integration points. These teams may or may not be on the same ART.

Team organization may evolve. Early product development often requires more innovation and tighter feedback (example #1 in Figure 2). Later development requires less creation and focuses on finalizing their parts for the overall product designs (examples #2 and #3 in Figure 2).

Agile teams capitalize on cross-functional skills to achieve the most value for each

iteration. A mechanical engineer may make simple firmware updates, or a software engineer may modify a CAD design and rerun analysis tests. Unfortunately, many organizations incentivize deep functional skills. An Agile transformation changes these incentives to encourage broader, cross-functional skills while not sacrificing the expertise required to build innovative products.

Assume Variability and Preserve Options

Traditional engineering begins new product development by specifying a product concept in precise detail in advance and then creating a detailed schedule to build it. However, history has shown this process to be unsuccessful, particularly for innovative products with many unknowns. Agile takes a different approach.

Samuel Langley and the Wright Brothers, leading competing teams in the race for flight, provide examples of these two approaches. After showing early success with his unmanned Aerodrome, Langley was well-funded. He then devoted years to perfecting a single design, which, unfortunately, contained fundamental flaws that would prevent it from flying.

In contrast, the Wright Brothers did not set out to design an airplane. Instead, they set out to close three knowledge gaps necessary for flight: lift, control, and propulsion. They shifted learning to the left through experiments and integrated prototypes to resolve those gaps.

The Wright Brothers were perhaps the first innovation team to apply [Set-Based Design](#). They tested repeated prototypes of subsystems (wings, rudder, elevators) that they frequently integrated with kites, gliders, and a flying machine. The integrations validated their decisions and provided fast feedback to make adjustments. They made final design decisions only after gaining sufficient knowledge and reducing the cone of uncertainty (Figure 3). For more detail, see SAFe [Principle#3 D Assume variability; preserve options](#).

Figure 3. Explore design alternatives to turn questions into knowledge

Figure 3. Explore design alternatives to turn questions into knowledge

As teams create knowledge, these decisions need to be captured and managed. In SAFe, teams store and modify this information in the [Solution Intent](#), the repository of system knowledge that includes the system's specifications. Being agile means these specifications are not fixed; they evolve based on learning. (Figure 4) The solution roadmap defines the learning milestones that drive the exploration activities and create the knowledge necessary to build the system. Every increment of work contributes new insights for the solution intent and moves specifications from variable to fixed.

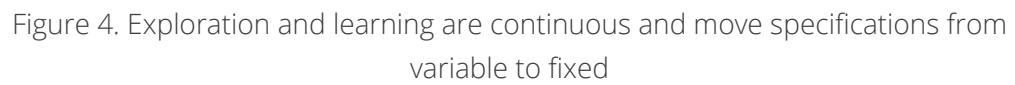


Figure 4. Exploration and learning are continuous and move specifications from variable to fixed

Build Incrementally, Integrate Frequently

Adhering to a fixed schedule based on phase-gate milestones in traditional development often defines success. Unfortunately, this approach forces decisions too early and can create false-positive feasibility [4]. In SAFe, milestones are based on objectively evaluating working systems ([SAFe Principle #5](#)). This requires all teams, including hardware, to frequently integrate their incremental changes into the solution.

Figure 5 contrasts the traditional approach with a continuous one. SpaceX strives to launch the next rocket version every 2-3 months to add new knowledge and data about the future solution. Learning is the goal, even when tests may seem like a failure (see Musk's tweet in Figure 5). Instead of focusing on qualifying every component (and subassembly and assembly), SpaceX focuses on creating the infrastructure to test the next version of the solution quickly. The transaction costs

for building and launching the next rocket are less than the holding costs of delayed learning. Watch [5] to see NASA's top Administrator, Jim Bridenstine, discuss what NASA has learned from working with SpaceX.

Figure 5. Contrast traditional and continuous development

Figure 5. Contrast traditional and continuous development

SAFe's roadmap and demonstrable milestones replace fixed schedules and phase-gate milestones. They define the future knowledge goals (for example, the wings can provide sufficient lift) that drive the teams' incremental work (for example, testing multiple wing designs in the wind tunnel and validating lift with the kite), as shown in Figure 6.



Figure 6. A SAFe perspective of the Wright Brothers' Flying Machine development

Integration points control product development [4] by creating knowledge from uncertainty (SAFe Principle #4) in large system development. ARTs integrate their teams' incremental changes so that the entire system is learning. And all teams and trains are on a common cadence, which creates natural integration points for the whole system.

Design for Change

The system must reasonably and quickly change in development and operational environments to support frequent integration. Modular designs that integrate through managed interfaces enable this. For example, Figure 7 shows a camera with an electrical interface to a vehicle control unit and a physical interface with the vehicle body. These components can evolve independently if they adhere to the interface specifications. Hardware interface specifications define properties for mechanical parts (size, weight, forces exerted, mounting) and electrical

components (connector type, pinouts, voltage).

Figure 7. Modular designs enable frequent, independent design iterations

While designing for change is not new, a poor understanding of economics can prevent good design decisions. For example, application-specific integrated circuit (ASIC) parts are often chosen because of their lower unit costs and power consumption. But their functionality is fixed; any change requires manufacturing and installing a new part. On the other hand, field-programmable gate arrays (FPGA) and system-on-a-chip (SOC) designs can be modified quickly and cost-effectively in both the development and operational environments. Physical joins often use solders or welds to reduce assembly costs. Non-permanent joins, like fasteners and connectors, add cost but make changes possible and more accessible. And some are now using additive manufacturing to create extensive subsystems that require no assembly. The economics must include the total costs for change and the value of evolving systems in the operational and development environments over the product's lifetime.

Perform Work in Small Batches

Agile software teams work in the context of Stories, small, vertical slices of functionality sized to be completed in a single iteration. Hardware engineers, and most individuals new to Agile, often struggle to break down work. However, this is not debatable or optional. Breaking work into small chunks is essential for frequent integration, fast experimentation, and the ability to adapt.

Twenty years ago, the software community had this same struggle when they

began their Agile journey. Over time, they created innovations (for example, microservices and API management) and tools for [Continuous Integration](#) and [Continuous Deployment](#) that simplified small batches of work. Today, most software developers can implement small, vertically-sliced stories and create working products every iteration. The hardware community is just beginning its Agile journey, which may be challenging. But in time, it will develop its practices, driving innovations and changes in its electrical and mechanical Computer-Aided Design (CAD) products that simplify small work.

Hardware developers can approach breaking work into small units of value in several ways. The first is considering the work required to reduce knowledge gaps for near-term milestones on the solution's roadmap.

The second addresses the work of integrating changes with a more comprehensive solution. For example, instead of designing an entire circuit board or mechanical part, create a few circuits or pieces of the part. Then, find a way to combine it with other system parts for feedback by integrating models, using breadboards, or joining 3D-printed parts. Engineering labs are filled with these types of experiments. Track this kind of work in backlogs and create a proactive mindset regarding small engineering changes and verification in the larger system context. Figure 8 shows an example Kanban board of stories for the camera team discussed earlier.

Figure 8. An example hardware team's Kanban board

Figure 8. An example hardware team's Kanban board

Hardware teams benefit from using well-defined Stories in other ways, particularly the Definition of Ready (DoR) and Done (DoD). This is particularly true for cross-functional teams, where members may be working in a relatively new domain, unaware of all the governance practices. The DoD defines the team's agreement on when work is complete and helps build quality.

A hardware DoD may include properly documenting data results and restoring equipment to a reusable state. The DoR is used less frequently by the Agile community and, thus, is not found in SAFe. However, it is helpful for hardware work that often depends on external items, such as equipment availability, support roles (such as only a safety engineer can approve power on), ordered material, and many others. A DoR communicates what is required to make an upcoming story actionable. The podcast in [6] describes DoR and DoD's value to Tesla's dynamic, cross-functional hardware teams, where iterations are measured in hours, not days.

Continuous Integration for Hardware Development

Developer changes are not verified until they can be integrated into the larger system context. Hardware development creates physical parts with material costs and long lead times. As a result, hardware verification often occurs later in the product development lifecycle, often near the end. Many strategies exist for shifting learning left at three stages of hardware development described below (Figure 9).

Figure 9. Continuous integration for hardware occurs at three locations

Figure 9. Continuous integration for hardware occurs at three locations

Virtual World ▶ Hardware development begins with models in design tools (for example, electrical and mechanical CAD). Many organizations now integrate virtual models to enable analysis and simulation for early feedback in a larger system context. Developers can verify their designs' compatibility with stakeholders before manufacturing parts.

Physical World ▶ Some design aspects can only be verified with physical parts. Many organizations already leverage additive manufacturing and 3D printing for early

feedback. They can also be used for production parts, mainly when innovation requires higher change frequency and low production volumes [1]. Reducing quality and reliability constraints on *development* parts can also accelerate manufacturing times. Development parts often operate in a controlled environment and are replaced by the next revision, unlike production parts which must last years in harsh environments.

Operational World Systems must be designed to enable easy changes in the working environment. The design choices discussed earlier (FPGAs over ASICs and connectors over solder) provide examples of design choices. Allocation of system functionality to design elements also plays a critical role. As network capacity and costs enable over-the-air updates, more behavior is allocated to software and programmable components.

Continuous Integration is taking small developer changes and testing, integrating, and validating them in a context ready for delivery. In hardware development, new functionality flows from virtual designs to physical parts that are then made available for easy installation in the operational environment.

Automating continuous integration activities accelerates the flow through these environments and provides faster feedback on developer changes. The continuous integration environment is a system, too, and developers must invest the time and resources to build and make the system. As Elon Musk says, ‘the factory is the product.’

Learn More

[1] SpaceX Triggers Increased Use Of 3D Printing, 2020

<https://www.fabbaloo.com/blog/2020/6/9/spacex-triggers-increased-use-of-3d-printing>

[2] Automakers use virtual reality to reduce development time for vehicles like the Hummer EV., 2020. <https://www.cnbc.com/2020/11/08/automakers-use-virtual-reality-to-cut-the-development-time-for-vehicles-like-the-hummer-ev.html>

[3] Why Tech Giants Like Amazon Are Designing Their Chips Ñ And Who Benefits, 2018. <https://www.thestreet.com/opinion/why-tech-giants-are-designing-their-own-chips-14807638>

[4] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development.*

Amacom, 2010.

[5] *Exclusive interview with Elon Musk and Jim Bridenstine about #DM2, SpaceX flies astronauts for the first time*, 2020. <https://youtu.be/p4ZLysa9Qqg?t=266>

[6] *Agile at Tesla with Joe Justice*. The Agile Wire Podcast, 2021.
<https://www.theagilewire.com/recordings/agile-at-tesla-with-joe-justice>

[7] *Inside Elon Musk's plan to build one starship a week*. Ars Technica, 2020.
<https://arstechnica.com/science/2020/03/insMusk's-on-musks-plan-to-build-one-starship-a-week-and-settle-mars>

Last update: 6 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)
[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)

English

Scaled Agile Framework



While we must acknowledge emergence in design and system development, a little planning can avoid much waste.

~James O. Coplien, Lean Architecture

Agile Architecture

Adjust Cookie Settings

Note: This article is part of [Extended SAFe Guidance](#) and represents official

SAFe content that cannot be accessed directly from the Big Picture.

Agile Architecture is a set of values, practices, and collaborations that support a system's active, evolutionary design and architecture.

This approach embraces the DevOps mindset, allowing the architecture to evolve continuously while supporting current users' needs. It avoids the overhead and delays associated with the **start-stop-start** nature and large-scale redesign inherent in phase-gate processes and Big Design Up Front (BDUF).

Agile architecture supports Agile development practices through collaboration, design simplicity, and balancing intentional and emergent design. It enables designing for testability, deployability, and releaseability, supported by rapid prototyping, domain modeling, and decentralized innovation.

Details

The architecture of a system can either accelerate or impede the ability to provide frequent, independent releases that allow the business to meet its objectives. Agile architects support business alignment by optimizing the architecture to support the value stream end-to-end. This optimization enables the company to achieve its goal of continually delivering value in the shortest sustainable lead time. Agile architects lead this process by supporting just enough [Architectural Runway](#) to support evolving business needs. They continually invest in legacy modernization initiatives and identify where to refactor, eliminating bottlenecks. Architects communicate the need for these ongoing technical objectives in clear business terms.

To support the continuous flow of value through the [Continuous Delivery Pipeline](#), Agile architecture:

- Evolves while supporting the needs of current users
- Avoids overhead and delays associated with phase-gate and BDUF methods
- Ensures the system always runs
- Balances emergent and intentional design
- Takes a systems view across the entire value stream

SAFe's [Lean-Agile Principles](#) inform Agile architecture practices. While all apply, three are particularly relevant to architecture. Before committing to a specific design, Agile architects use fast learning cycles ([Principle #4](#)) to explore alternatives ([Principle #3](#)) and arrive at the optimal solution. They also create the environment

for decentralized decision-making ([Principle #9](#)) by defining and communicating the architectural vision and strategy and then collaborating with and coaching the teams who build it.

Collaborating Design with Architect Roles

SAFe defines three architect roles: [Enterprise](#), [Solution](#), and [System](#), which address these concerns at their respective levels ([Portfolio](#), [Large Solution](#), and [Essential](#)). These architects regularly collaborate across and among levels to ensure alignment and address issues and concerns as they arise. Figure 1 illustrates all the skills architects need to make technical decisions. It's often the case that more than one architect is required to ensure sufficient knowledge and prevent architecture decisions from bottlenecking teams.

Figure 1. SAFe architecture roles span architectural domains

Figure 1. SAFe architecture roles span architectural domains

The interdependent nature of business and technical strategy requires

collaboration between architects and other SAFe roles to ensure that the architecture meets the organization's and its customer's current and evolving needs. Within the [Agile Release Train](#) (ART), System Architects communicate the technical path through the architectural runway, [Nonfunctional Requirements](#), and the design and support of the [Continuous Delivery Pipeline](#) (CDP). Architecture also enables [Built-in Quality](#). The [Systems Team](#) helps realize the architectural vision by building the supporting infrastructure that allows [Agile Teams](#) to design, implement, test, and deliver value. System Architects coordinate with Enterprise and Solution Architects to ensure their solutions are aligned with the larger vision. Finally, architects in any role are also [Lean-Agile Leaders](#) responsible for mentoring teams and enhancing the overall capabilities of teams.

Balancing Intentional and Emergent Design

Traditional architecture approaches led to extensive early architecture work. This can create overly comprehensive documentation and unvalidated decisions. An alternative approach is designing the architecture to better align with the business needs, which requires the following combination of intentional and emergent design:

Intentional architecture ☰ Defines a set of purposeful, planned architectural strategies and initiatives that enhance solution design, performance, and usability, guiding cross-team design and synchronized implementation.

Emergent design ☰ Provides the technical basis for a fully evolutionary and incremental implementation approach. This helps developers and designers respond to immediate user needs, allowing the structure to evolve as the system is built and deployed.

With this balance, Agile architecture is a Lean-Agile approach to addressing the complexity of building enterprise solutions. It supports the needs of current users while evolving the system to meet near-term future needs. Used together, emergent and intentional design continuously develop and extend the architectural runway that provides the technical foundation for the future development of business value.

To implement a balanced approach to design, architects, and teams should prioritize the following:

1. Understand the system's goals and requirements before designing the system
2. Identify the areas where intentional design is necessary to build a solid foundation
3. Be willing to adapt and evolve the design as the system is developed
4. Encourage experimentation and learning to discover new and innovative solutions
5. Strive for consistency and cohesion in the design to maintain the system's quality

By balancing intentional and emergent design, the team can build a flexible, adaptable, and responsive system to changing business needs while maintaining a clear direction and vision for future development.

Architecting for DevOps and Release on Demand

Agile architecture fosters a DevOps culture by ensuring [Solutions](#) are architected for continuous delivery. Architects participate in the design and execution of the CDP and evangelize and exemplify SAFe's CALMR principles (see the [DevOps](#) article for a complete description of SAFe CALMR principles). They allow decisions to emerge by defining minimum viable (just enough) architecture, ensuring loose coupling between system elements, supporting the creation and evolution of interfaces, and fostering architecture as code through standard annotations, attributes, and naming conventions. Agile architecture also provides built-in quality by automating architectural compliance tests.

To support continuous deployment, Agile architecture decouples deployment from release. Functionality is *continuously deployed to a production environment* but only available to end-users on demand. Deploying frequently builds trust in the CDP pipeline and reduces delays caused by more traditional release governance practices. This trust enables individual teams and ARTs to explore and test ideas in an actual production environment independently.

The CDP begins and ends with a benefit hypothesis, defining it in [Continuous Exploration](#) and eventually measuring it in [Release on Demand](#). The system architecture provides the telemetry to measure hypotheses to support [innovation accounting](#) and other usage data for teams and ARTs to validate their assumptions. Agile architecture supports the CDP by considering other system factors as critical architectural concerns, such as test architecture and test data management.

Aligning Architecture with Business Value

In the digital age, businesses rely on technology to deliver value to their customers. As business strategy changes, the technology, systems, and business applications that provide it must change. Figure 2 shows an example [Operational Value Stream](#) for a customer order and product delivery. It illustrates the operational steps with the systems and applications that support them below. Those supporting applications and systems realize any business changes to the customer

experience. Architects work closely with [Business Owners](#) and [Product Managers](#) to ensure those systems can recognize current and future business goals.

Figure 2. Systems in support of a customer's order placement and delivery

Figure 2. Systems in support of a customer's order placement and delivery

[Strategic Themes](#) and the [Portfolio Vision](#) influence architecture and drive the architecture runway. They provide the constraints, direction, and overall context for technology investments in a portfolio. Looking more broadly, architecture must also consider the larger enterprise strategy, including awareness across portfolios, especially for [Enterprise Architects](#).

Developing Solution Vision, Intent, and Roadmaps

Aligning architecture with business strategy accelerates achieving business goals. Architects realize these goals by translating strategic themes into solutions. These

solutions are defined by their [Vision](#), [Solution Context](#), and [Solution Intent](#). The solution intent includes the decisions, patterns, models, and additional technical information to serve as minimally sufficient documentation, such as design constraints, including [Nonfunctional Requirements](#) (NFRs).

[Roadmaps](#) define a plan to realize the solution. Architects and teams collaboratively define [Enablers](#) in the roadmap to support exploring technical options and building the architectural runway, providing early feedback on achieving those milestones. Teams provide feedback on architectural decisions as they implement features balancing intentionality and emergence.

The roadmap drives backlogs, which define work for an ART. Architects collaborate with [Product Management](#) on prioritizing and balancing new functionality with technical work. They anticipate technical debt and help implement [Principle #6](#)—Make value flow without interruptions, advocating changes to the architectural runway and ensuring they have high priority.

Prioritizing Features and Enablers for PI Planning

With each increment, teams build the highest priority [Features](#) and enablers. Architects collaborate with Product Management to define and prioritize these near-term work items. They provide feasibility insights, helping define and scope current features with acceptance criteria. They also consider future features and define enablers in the backlog for teams to explore and gain knowledge that ensures viability.

Architects also consider technical dependencies with other ARTs or Solution Trains, acting as critical collaborators in these [Coordination](#) activities. They collaborate with teams to reduce dependencies during PI Planning, helping teams make the necessary decisions and adjustments.

Coordinating Agile Architecture in PI Planning

During PI Planning, architects support the teams by presenting the architectural briefing as part of the planning agenda. As teams make their plans during breakouts, architects help ensure they plan technical work properly and account for the ART's enabler work. And they address any questions and concerns.

Architects support the management review to address architectural and technical

issues on potential adjustments. They also participate with [Business Owners](#) as they assign value to the teams' PI Objectives. They explain, in business terms, how enablers and other technical work support their overall objectives and lobby for their need and importance.

Supporting Continuous Delivery

Adopting Agile architecture is critical to support ARTs and Solution Trains to implement technical and exploration with enablers, and, as such, architects often guide teams on their execution. For example, architects may attend various teams' Iteration Planning and System Demos to review architecture progress, address issues, and adjust direction. They are also generally available to the teams for coaching and mentoring, ensuring problems and issues are addressed quickly so that architecture is not a bottleneck.

With each increment, architects ensure teams balance *intentional* and *emergent* design by reviewing the results of enabler work, including new knowledge, additions to the architecture runway, and CDP. For Large Solutions, architects stay aligned and share progress at the *Architect Sync* event shown in Figure 3.

Figure 3. Architect Sync during the Solution Train PI execution

Figure 3. Architect Sync occurs during the Solution Train PI execution

Supporting New Strategic Themes and Value Streams

Architecture must evolve to meet changing business needs and opportunities. Otherwise, technology becomes the bottleneck for business execution. Changes in business strategy are reflected in new or modified strategic themes, which, through the [Portfolio Canvas](#), translate into new or modified solutions and value streams. Enterprise Architects support and influence this process by providing input, attending Value Stream Mapping workshops, and setting expectations on technical feasibility. Once the new direction is made, Enterprise Architects collaborate with System and Solution architects to realize the new business direction. They communicate the new strategy and show how it changes the solution vision, intent, and roadmap.

Enterprise Architects coordinate architectural work across the portfolio, ensuring alignment across solutions and value streams. They provide technical guidance for the long-term evolution of the technologies and platforms and the more extensive nonfunctional requirements (security, compliance, performance, and more) for the portfolio solution set. And they often serve as [Epic Owners](#) for portfolio-level Enablers to ensure significant technological shifts remain in line with business strategy.

Leading the Lean-Agile Transformation

Due to their knowledge and experience, architects are often respected and highly regarded by the development community. Therefore, architects play a vital role in any SAFe transformation. Architects are [Lean-Agile Leaders](#) and, as such, model leaner ways of thinking and operating, so developers learn from their example, coaching, and encouragement. They enable autonomy and encourage mastery to grow the development community's knowledge base and skill set. SAFe architects embody the new way of working, participate in creating the organization's [\(Implementation\) Roadmap](#), and help accelerate the adoption as Lean-Agile leaders.

Learn More

[1] Manifesto for Agile Software Development. <http://AgileManifesto.org/>

[2] Crispin, Lisa, and Janet Gregory. *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison-Wesley, 2009.

[3] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[4] Gregory, Janet, and Lisa Crispin. *More Agile Testing: Learning Journeys for the Whole Team*. Addison-Wesley, 2015.

Last update: 27 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework

Extended SAFe Guidance

The Extended SAFe Guidance page provides official content from Scaled Agile, Inc. that cannot be accessed directly from the [SAFe Big Picture](#). These articles are an integral part of the SAFe knowledge base.

What's New
in SAFe 6.0

Adjust Cookie Settings 

[Agile Architecture in SAFe](#)

[Agile Contracts](#)

[Agile Software Engineering Landing Page](#)

[Agile Testing](#)

[Agile Workspaces](#)

[Applying Kanban in SAFe](#)

[Applying SAFe to Hardware Development](#)

B

[Balancing the Dual Operating System](#)

[Behavior-Driven Development](#)

C

[CapEx and OpEx](#)

[Complying with Regulatory and Industry Standards](#)

D

[Distributed PI Planning](#)

[Domain Modeling](#)

E

[Evolving Role of Managers in Lean-Agile Development](#)

F

[Facilitating SAFe Assessments](#)

I

[Innovation Accounting in SAFe](#)

O

[Organizing Agile Teams and ARTs: Team Topologies at Scale](#)

R

[Recommended Reading](#)

[Refactoring](#)

S

[SAFe for Marketing](#)
[SAFe Requirements Model](#)

T

[Test-Driven Development](#)

U

[User Story Primer White Paper](#)

W

[Working Successfully in Agile with Remote Team Members](#)
[What's new in SAFe 6.0?](#)

Last update: 14 March 2023

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework

SUBSCRIBE TO THE SAFe
BLOG

Say Hello to SAFe 6.0!

NEWS

March 15, 2023 [Leave A Comment](#)

By [Dean Leffingwell](#)

CATEGORIES

[NEWS \(3\)](#)

Our mission at Scaled Agile has always been to help those who build the world's most essential systems do so better, faster, at less cost—and with more enjoyment in the process. It's self-evident that a mission like that has no end.

[SAFE FRAMEWORK UPDATES \(1\)](#)

Especially now, as the unprecedented times of the pandemic, global economy, and supply chain issues have driven home a reality that has been decades in the making

CALENDAR

APRIL 2023

Enterprises are overwhelmed by constant change. And while all this is happening, advanced technologies such as AI, big data, and the cloud radically transform how we innovate. The complexity and criticality of these new systems have never been higher. Indeed, the general welfare of humanity—and sometimes our very lives—literally depends on them.

As we learn new things from our customers, partners, and market trends, we constantly evolve the Framework with new and updated guidance. Occasionally, we take a critical look at the entire Framework—from the SAFe Big Picture to all of the supporting knowledge base articles, graphics, and courseware—and decide it's time for a new version. One that will ensure our customers continue to get the guidance and resources they need to adapt to market shifts and transform disruption into opportunity.

To that end, we are excited to welcome you to SAFe™ 6.0, the latest version of the Framework, courses, certifications, toolkits, and online learning.

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Ç Mar

RECENT POSTS

SAY HELLO TO SAFE 6.0!

MARCH 15, 2023

FRAMEWORK PROMOTIONS

MARCH 6, 2023

EXCITING NEW BOOK: SAFE FOR

DEVOPS PRACTITIONERS

MARCH 6, 2023

TRAINING

COURSE CALENDAR

ABOUT CERTIFICATION

BECOME A TRAINER

SAFe 6.0 Release Themes

This release represents a significant advance in how enterprises integrate SAFe practices in day-to-day work, make the change stick, and achieve the benefits of true business agility. SAFe 6.0 includes many new and advanced practices to support and enable the latest technology and business trends, including eight ways to

accelerate the flow of value and guidance for expanding SAFe across the organization into other business functions.

Six release themes highlight the most relevant changes and additions:

1. **Strengthening the foundation for Business Agility**

ÑThe fundamental aspects of business agility, including the [Business Agility](#) Value Stream (BAVS) and foundation articles, are substantially improved with a new, leaner mindset and updated core values. Updates to the SAFe Implementation Roadmap and revised responsibilities for [SAFe Practice Consultants](#) (SPCs) clarify the path for navigating the change.

2. **Empowering teams and clarifying roles and responsibilities**

ÑNew guidance helps individuals better understand their roles and improve job performance,

leading to greater support for the organization's goals and increased personal satisfaction. An example of the new role— responsibility wheel— for the newly titled [Scrum Master/Team Coach](#) is shown below.

3. **Accelerating Value Flow** A deeper understanding of achieving flow is central to this evolution of SAFe. Our latest guidance defines eight flow accelerators (as shown below) designed to improve flow throughout SAFe. And with our ability to measure flow, we have a new and quantitative basis for understanding what's happening, how things are working, and what we can do to improve. New SAFe Scrum, SAFe Team Kanban, Built-in Quality, and Value Stream Management articles add flow directly into the teams' daily work. Four new flow articles provide advanced guidance on how to apply [Principle 6 D Make value flow without interruptions](#) to Agile Teams, ARTs, Solution Trains, and Portfolios.

This may be the most significant breakthrough in SAFe and our industries' understanding of measuring and improving what was formerly intangible and immeasurable.

4. **Enhancing Business Agility with SAFe across the business**—The new [Business and Technology](#) article highlights five proven patterns that can be used to extend agility throughout the enterprise. These include:

5. **Building the future with AI, Big Data, and Cloud**—SAFe 6.0 helps accelerate the adoption of the latest advanced technologies through expanded guidance for integrating [AI](#), [Big Data](#), and [Cloud](#) into value streams. Applying these technologies has become critical for competing now and in the future.

6. Delivering better outcomes with Measure and Grow

and OKRs

SAFe 6.0 helps enterprises provide better business outcomes with expanded guidance on applying [Objectives and Key Results \(OKRs\)](#) and improved techniques for measuring and enhancing competency and flow.

Updated Learning and Practices Resources Studio

And, of course, this latest guidance is reflected in updates to SAFe courseware, online learning, and practice resources. Leading SAFe™, one of Scaled Agile's most popular courses, is now available in languages including Brazilian Portuguese, Chinese, French, German, Japanese, Korean, and Spanish as part of our ambitious and ongoing localization roadmap.

SAFe™ Studio

As a part of this release, we have launched SAFe™ Studio, our new subscription-based platform that enables SAFe professionals to learn, practice, and manage SAFe. It helps translate guidance into execution, helping employees adopt and excel in a different way of working throughout all the business units in the enterprise.

Want to learn more?

This is just a small appetizer of what's new in SAFe 6.0. To get the full story, please read the [What's new in SAFe 6.0 article](#). And be sure to watch the [SAFe 6.0 launch announcement videos](#) to learn more about SAFe Studio, SAFe 6.0, and other exciting new products to support you on your SAFe journey.

Welcome to 6.0!

Finally, we owe a special thanks to all of you in the SAFe community for your feedback and support. Your passion for the best SAFe possible has driven us to deliver what we like to believe to be the most capable and adaptable Framework in the world.

Your Framework Team
Dean Leffingwell, Andrew Sales, Dr. Steve Mayner, Dr. Harry Koehnemann, Marc Rix, Rebecca Davis, and Cheryl Crupi, with special contributors Richard Knaster and Alex Yakyma.

 *Share Post*

AUTHOR INFO

DEAN LEFFINGWELL

Recognized as the one of the world's foremost authorities on Lean-Agile best practices, Dean Leffingwell is an author, entrepreneur, and software development methodologist.

[Leave a Reply](#)

PREV POST

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework

[SUBSCRIBE TO THE SAFE
BLOG](#)

Framework Promotions

NEWS

SAFE FRAMEWORK UPDATES

March 6, 2023

[Leave A Comment](#)

By [Dean Leffingwell](#)

CATEGORIES

[NEWS \(3\)](#)

Hi Folks,

[SAFE FRAMEWORK UPDATES \(1\)](#)

One of the coolest things about our mission at Scaled Agile is that the challenge of Ôhelping people build the worldÕs most important systemsÕ lays ahead of us and not behind.

CALENDAR

As we look into the future, we see our customer needsÑ
Adjust Cookie Settings and the complexity of the systems they buildÑgrowing,

ABRIL 2023

putting even greater demands on SAFe. To meet these demands, we are evolving the Framework team.

My teammates are some of the smartest and wisest people I know so I couldn't be happier to announce these promotions:

Andrew Sales will assume the title and responsibilities of Chief Methodologist alongside me. In addition, he will continue as Product Manager of the Framework.

Dr. Stephen Mayner will assume the title and responsibilities of Methodologist and Framework VP. To provide additional thought leadership, emphasis, and critical connections to the community, partners, and enterprises, **Harry Koehnemann, Marc Rix, Rebecca Davis**, and **Cheryl Crupi** will assume the title and responsibilities of Methodologists.

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Ç Mar

RECENT POSTS

SAY HELLO TO SAFE 6.0!

MARCH 15, 2023

FRAMEWORK PROMOTIONS

MARCH 6, 2023

EXCITING NEW BOOK: SAFE FOR DEVOPS PRACTITIONERS

MARCH 6, 2023

TRAINING

Please join me in congratulating this extraordinary group of people. I can't imagine a better team to guide the Framework into a fun and productive future. It is rewarding and fulfilling to continue to be a part of it.

Stay SAFe,

ÑDean

COURSE CALENDAR

ABOUT CERTIFICATION

BECOME A TRAINER

Share Post

AUTHOR INFO

DEAN LEFFINGWELL

Recognized as the one of the world's foremost authorities on Lean-Agile best practices, Dean Leffingwell is an author, entrepreneur, and software development methodologist.

Leave a Reply

[PREV POST](#)

[NEXT POST](#)

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework

SUBSCRIBE TO THE SAFe
BLOG

Exciting new book: SAFe for DevOps Practitioners

CATEGORIES

NEWS

March 6, 2023 [Leave A Comment](#)

NEWS (3)

By [Harry Koehnemann](#)

SAFE FRAMEWORK UPDATES (1)

If you attended a DevOps conference several years ago, you would have seen sessions filled with deep discussions on technical, architectural, and coding practices. Over time, the DevOps community has increased interest in other areas, including Lean-Agile methods, leadership, and organizational change. Indeed, for DevOps to succeed, the tooling and technical practices must integrate with the new ways of working. Organizations need to think of them

CALENDAR

APRIL 2023

holistically as we do in SAFe.

In his new book, *SAFe for DevOps Practitioners*, Bob Wren connects the SAFe principles and practices enabling business agility to the technical methods and tooling espoused by DevOps. What grabbed my attention is how Wren structured the book around SAFe's CALMR and Continuous Delivery Pipeline (CDP) structures and provides a deep dive into the various DevOps practices and tools that enable them.

SAFe practitioners will connect with the book's organizing structure around CALMR and the Continuous Delivery Pipeline (CDP) and see how relevant DevOps technical tooling and techniques apply. And we can use it to show DevOps practitioners how their technical and tooling practices align with SAFe's principles and practices to enable business agility across the enterprise.

While the book can be read from cover to cover, I think it will also be useful as a reference. A reader familiar with SAFe's CALMR and CDP organizing structures will find value in reading individual sections to connect SAFe topics to the appropriate DevOps tooling and practices.

Practitioners in the SAFe and DevOps communities help enterprises build some of the world's most important systems. SAFe change agents enable business agility in most of the Global 2000, helping those enterprises survive and thrive in the digital age. DevOps practitioners in these same organizations provide the tooling and technical processes that deliver digital products faster and more reliably. *SAFe for DevOps Practitioners* shows how SAFe change agents and DevOps practitioners work together to

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Ç Mar

RECENT POSTS

SAY HELLO TO SAFE 6.0!

MARCH 15, 2023

FRAMEWORK PROMOTIONS

MARCH 6, 2023

EXCITING NEW BOOK: SAFE FOR DEVOPS PRACTITIONERS

MARCH 6, 2023

TRAINING

COURSE CALENDAR

ABOUT CERTIFICATION

BECOME A TRAINER

accelerate business success.

I hope you find this new book as useful in your SAFe transformations as I think you will. Check it out in our [Books section](#), including a link to purchase options on Amazon.

Stay SAFe!

Harry

 *Share Post*

AUTHOR INFO

HARRY KOEHNEMANN

[Leave a Reply](#)

NEXT POST

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)

English

Scaled Agile Framework

Download the Big Picture

Subject to the important notice below, you can download a copy of the SAFe Big Picture in multiple formats. There are four configurations, each designed to meet the needs of every possible implementation scenario, from the most basic to the most complex. Not sure which one is right for you? [Click here to learn more.](#)

Choose from four configurations

Adjust Cookie Settings [Full SAFe](#)

[Portfolio SAFe](#)

[Large Solution
SAFe](#)

[Essential SAFe](#)

<p>Full SAFe represents the most comprehensive configuration. It supports building large, integrated solutions that typically require hundreds of people or more to develop and maintain.</p> <p>Download 8.5 x 11 Download 48 x 60 Download A4</p>	<p>Portfolio SAFe is for enterprises building solutions that require a modest number of Agile teams. It supports the development of multiple solutions, which have minimal dependencies on one another.</p> <p>Download 8.5 x 11 Download 48 x 60 Download A4</p>	<p>Large Solution SAFe is for enterprises that are building large and complex solutions, which do not require the constructs of the portfolio level.</p> <p>Download 8.5 x 11 Download 48 x 60 Download A4</p>	<p>Essential SAFe is most basic configuration of the framework and it provides the minimal elements necessary to succeed with SAFe.</p> <p>Download 8.5 x 11 Download 48 x 60 Download A4</p>
---	---	--	---

Download the SAFe Overview

The SAFe Overview provides a simplified view of the Framework's Seven Core Competencies of Business Agility and their twenty-one dimensions that enable business agility. The execution-related competencies are shown on the left, while the competencies that support strategy development are on the right. The Lean-Agile Leadership competency which is the foundation is in the bottom middle. The customer is prominently featured at the center as the focal point for all the competencies. Measure and Grow at the top right is a reminder of the importance of periodic self-assessments to track the organization's progress towards the

principles and practices that enable business agility.

This overview is a useful tool for providing an initial orientation to SAFe, as well as an introduction to the business agility assessment, and for framing conversations with senior leaders.

[Download 8.5 x 11](#)

[Download 48 x 60](#)

[Download A4](#)

Download the Implementation Roadmap

The SAFe Implementation Roadmap overviews a strategy and an ordered set of activities that have proven to be effective in successfully implementing SAFe.

Based on proven organizational change management strategies, the SAFe Implementation Roadmap overviews the steps, or “critical moves,” an enterprise can take to implement SAFe in an orderly, reliable, and successful fashion.

[Download 8.5 x 11](#)

[Download 48 x 60](#)

[Download A4](#)

Download SAFe Posters

SAFe

Agile

Principles

The SAFe Principles poster lists ten immutable, underlying Lean and Agile principles upon which SAFe is based. These are the fundamental tenets, basic truths, and economic underpinnings that drive the roles and practices that make SAFe effective.

Manifesto

Agile development is the heart of SAFe. While there is no one perfect definition, the Agile Manifesto is the foundational document. Its four values and twelve principles have withstood the test of time and continue to be the touchstone by which true agility can be judged and recognized.

SAFe
Core
Values

SAFe
Flow

The four
Core Values
of

alignment, transparency, respect for people, and relentless improvement represent the foundational beliefs that are key to SAFe's effectiveness.

These tenets help guide the behaviors and actions of everyone participating in a SAFe portfolio. Those in positions of

Accelerators

Each flow property is subject to optimizations, and often many steps encounter unnecessary delays, bottlenecks, and other impediments to flow. *Making value flow without*

authority can help the rest of the organization embrace these ideals by exemplifying these values in their words and actions.

interruptions can best be achieved by adopting the eight Œflow acceleratorsŒ described in this article. These powerful accelerators of value are relevant to all Framework levels, but the challenges differ for each.

Order SAFe 6.0 Big Picture Travel Fabric Posters

Need a Big Picture in the field, but donŒt want to lug around a giant poster board? SAFe is available in a handy crease-resistant fabric that is so lightweight it will fit in your laptop case! Specifically formulated for high-resolution printing, the SAFe travel-fabric poster features crisp images and graphics.

The Fine Print

All copyrights in the above graphics remain the property of Scaled Agile, Inc., but we hereby grant permission to the person or entity downloading them to reproduce, distribute, perform, display, and use these graphics, free of

charge, on the condition that: (1) you do not compete with any product or training provided by or for Scaled Agile, Inc. or its licensees, and (2) you do not modify any of the graphics or remove any trademark or copyright. You do not have the right to create any derivative works based upon them without the express written permission of the copyright holder, Scaled Agile, Inc.

We make certain presentations and tools available for download under more permissive terms at [Presentations and Videos](#)

All other SAFe™ content and graphics are governed by these [usage restrictions](#)

Please refer to [Frequently Asked Questions](#) if you have questions about usages of other SAFe content

Any permission questions or requests can be submitted via the [Permission Request Form](#)

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

[Blog](#)

[Twitter](#)

[Usage and Permissions](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Linkedin](#)

[YouTube](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

[Scaled Agile Framework](#)

Presentations and Videos

We offer the following *Presentations and Videos* to help start or enhance your SAFe learning journey. You may also be interested in these additional resources: [SAFe Books](#), [Extended SAFe Guidance](#), and [Scaled Agile Resource Library](#).

SAFe Overview in 5
Minutes

Navigating the SAFe
6.0 Big Picture

[Adjust Cookie Settings](#)

Ideal for people just getting acquainted with the Framework, this 5-minute video provides a high-level overview of SAFe and the Seven Core Competencies of Business Agility.

Have you ever looked at the SAFe Big Picture and wondered where you should start? If so, watch this video to take a tour of the Big Picture and understand all its parts.

Why SAFe?

Why is SAFe™ the world’s most widely used framework for business agility? Get the answer from SAFe™ creator, Dean Leffingwell in this 18-minute video. Dean shares actual results achieved by some of the world’s leading companies including American Express, Bosch, Porsche, Lockheed Martin, Pepsico, and others.

You’ll learn about SAFe™’s core competencies and key practices that contribute to these results: mindset and culture, Lean-Agile leadership, Lean startup thinking, customer centricity, and design thinking. You’ll understand why one set of roles, responsibilities, and guidelines has proven to be so effective and how you can leverage the new [Business Agility value stream](#) in your own organization. Note: This keynote was recorded for the Agile Experience event in Portugal

SAFe 6.0 Brief Overview

In this short video, Chief Methodologist and Framework Product Manager Andrew Sales presents the six primary SAFe 6.0 release themes and highlights how SAFe Studio supports Enterprises to transform towards Business Agility.

in 2021.

What's New in SAFe™ 6.0

A hallmark of SAFe is that it stays current with new and evolving business and technology trends. Our incremental, continuous delivery model enables enterprises to adapt more quickly and stay ahead of the competition by leveraging the latest SAFe knowledge.

In this video, Chief Methodologist and Framework Product Manager Andrew Sales walks you through all the key changes in SAFe 6.0 and explains why they are critical to achieving Business Agility.

Download and review the PowerPoint for a detailed walkthrough of What's New in SAFe 6.0: Thriving in the Digital Age with Business Agility.

The presentation is organized around the six primary release themes of SAFe 6.0:

- Strengthening the foundation for Business Agility
- Empowering teams and clarifying responsibilities
- Accelerating value flow
- Enhancing Business Agility with SAFe across the business
- Building the future with AI, Big Data and Cloud
- Delivering better outcomes with measure and grow and OKRs

[Download](#)

Introducing SAFe 6.0

One of our most popular downloads, the Introducing SAFe PowerPoint is ideal for presentations to audiences

just learning about the Framework. Complete with speaker notes, the deck is designed to help you communicate SAFe's purpose, principles, and mindset, the business benefits that it can deliver, and the seven core competencies that support business agility.

[Download](#)

Topics included:

- Thriving in the digital age
- SAFe Lean-Agile Mindset, Core Values and SAFe Principles
- Accelerate time-to-market with the Business Agility Value Stream
- Seven Core Competencies of Business Agility
- Measuring and improving flow with flow metrics and the eight flow accelerators
- Lean, practice, and manage with SAFe Studio
- SAFe Courseware, SAFe Community, Global Partner Network and SAFe Enterprise Subscription
- Customer stories and business outcomes

Like the Framework itself, this material is copyrighted and usage guidance and restrictions apply (see Fine Print below for details).

SAFe Customer Stories

Read more [Customer Stories](#) to learn how some of the world's most successful enterprises are using SAFe

Accept

Accept [Advertisement](#)

Accept [Advertisement](#)

[Advertisement](#)
 cookies to view the content.

cookies to view the content.

cookies to view the content.

Designing the Digital Future at Porsche:
 SAFe Customer Keynote
 42:00

An Oracle Success Story: How a Mission-critical Cloud Migration Brought the Business and IT Together
 46:00

CVS Health #OneStepCloser:
 SAFe Customer Story
 34:00

The Fine Print

All copyrights in the above materials remain the property of Scaled Agile, Inc., but we hereby grant permission to the person or entity downloading them to reproduce, distribute, perform, display, and use all or any part of these presentations and tools, free of charge, on the condition that: (1) you do not compete with any product or training provided by or for Scaled Agile, Inc. or its licensees, and (2) you do not modify any of the original slides, or remove any trademark or copyright.

We make certain graphics available for download under more permissive terms at [Download SAFe Posters and Graphics](#) page.

All other SAFe™ content and graphics are governed by the usage restrictions at [Usage and Permissions](#).

Please refer to [Frequently Asked Questions](#) if you have questions about usages of other SAFe content.

Any questions or other requests can be submitted via the [Permission Request Form](#).

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework

Usage And Permissions

Usage Guidelines and Restrictions

Scaled Agile, Inc. is committed to helping ensure the success of those who apply the Scaled Agile Framework™ (SAFe™). To help achieve this, we've made SAFe knowledge publicly available throughout this website free of charge. However, all content, including text and graphics, is protected by U.S. and International copyright laws and may not be copied, modified to create derivative works, or distributed without our express written permission.

We make certain materials available for download under more permissive terms at [SAFe Posters & Graphics](#) and [Presentations and Videos](#).

Adjust Cookies Settings
Permissions

If you have a question regarding permissions or licensing of our content, please visit Scaled Agile's [Permissions FAQs](#) (includes access to a [Permissions Request form](#) if required).

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*You think that
because you
understand ÔoneÕ
that you must
therefore
understand ÔtwoÕ
because one and
one make two. But
you forget that you
must also
understand Ôand.Õ*

ÑDonella H.
Meadows

[Adjust Cookie Settings](#)

Solution Architect

Find a Course:

Go

The Solution Architect is responsible for defining and communicating a shared technical and architectural vision for a Solution Train to help ensure the solution under development will be fit for its intended purpose.

This article describes the role Solution Architects play in SAFe. It guides those building large-scale IT systems as well as those building large, cyber-physical, engineered systems. Many large systems—satellites, vehicles, robotics, medical devices, and more—have both cyber-physical and large-scale IT elements. In practice, the Solution Architect role is typically filled by a small team rather than one individual.

Details

Solution Architects play a critical role in the Enterprise Solution Delivery (ESD) core competency by aligning the many solution builders across multiple Agile Release Trains (ARTs) and Suppliers to a shared technical direction. To do this, they collaborate with various Solution Train roles and Agile Teams to elaborate the solution, validate technology assumptions, evaluate implementation alternatives, and converge on the final solution.

Key Collaborations

Designing a successful large solution is a highly collaborative process that requires ongoing communication, coordination, and cooperation across different people and teams involved in large solution development. This process is led by the Solution Architect and involves the key collaborations depicted in Figure 1.

Figure 1. Key collaborations of Solution Architect

Figure 1. Key collaborations of Solution Architect

The most critical interactions appear in the following areas:

Steer the Solution Train ☰ The Solution Train's forward momentum is guided by three key roles. The Solution Architect provides an architecture that supports the business direction. **Solution Management** determines the solution's direction; the **Solution Train Engineer** (STE) facilitates the solution development process. These roles intersect and collaborate at PI boundaries and during PI execution. The architect collaborates with the Solution Management to develop the **Solution Vision**, **Solution Roadmap**, and the **Capabilities** required to meet them. They also collaborate with the STE to eliminate the Solution Train's technology impediments and provide better synergy between the solution architecture and the Solution Train structure and process. As a critical role in steering the Solution Train, the architect participates in various aspects of **Coordinate and Deliver**.

Align on enterprise architecture ☰ In addition to delivering on its purpose for the intended customer, every large solution must also align with the requirements of the rest of the organization and business ecosystem. Much of this pertains to enterprise architecture and requires close interaction of the Solution Architect with the

Enterprise Architect. Some of these collaborations occur naturally in the context of the **Portfolio Kanban** system. Often, the Solution Architect may perform the role of the **Epic Owner** for an architectural epic in portfolio kanban. Aligning with the rest of the organization may also involve interactions with Solution Architects from other Solution Trains.

Evolve solution architecture Ⓛ With Agile, large solutions are not designed one time, top-down. Instead, the high-level architectural evolves in the context, reality, and constraints of what the trains can deliver. This involves close interactions between the Solution Architect, the **System Architects**, and the Suppliers within the Solution Train. Their continuous collaboration culminates at every PI boundary.

Work with special teams Ⓛ The Solution Architect works with the teams responsible for the different functions required to support the solution development work. The **System Team** is one such example. There may be a System Team at the Solution Train level, or select ART-level System Teams participate in the work critical to the solution. This often entails enabling solution integration and testing or providing the optimal development infrastructure. Additionally, the Solution Architect may be involved with **Shared Services** responsible for other concerns, such as deployment, system reliability engineering, user experience design, data management, compliance, etc.

Responsibilities

The Solution Architect's responsibilities can be categorized into the following responsibility areas, as shown in Figure 2.

Figure 2. Areas of responsibility of a Solution Architect

Figure 2. Areas of responsibility of a Solution Architect

Each of these responsibility areas is described in the sections below.

Implementing Lean Systems Engineering

The Architect has a crucial role in evolving the way of building large cyber-physical systems that usually involve a highly diverse technology stack and multiple ARTs and suppliers. The practices for implementing Lean Systems Engineering define how this is approached in [Enterprise Solution Delivery](#). For reference, the practices are as follows:

1. Specify the solution incrementally
2. Apply multiple planning horizons
3. Design for change
4. Frequently integrate the end-to-end solution

5. Continually address compliance concerns
6. Use Solution Trains to build large solutions
7. Manage the supply chain
8. Build an end-to-end Continuous Delivery Pipeline
9. Evolve deployed systems
10. Actively manage artificial intelligence/machine learning systems

The architect has a vital role in implementing these practices. Some of these are specific architect responsibilities that are discussed further in this article.

Establishing Solution Intent and Context

Successful solution development requires a clear understanding of what it is to be developed and delivered. The Solution Architect plays a crucial role in the solution definition providing the technology direction. This is achieved in an Agile manner where the Solution Intent and Context are progressively explored based on the learning milestones. The Solution Architect does the following:

Identify and respond to technology-enabled business opportunities ☰

Succeeding in the digital age requires much more than simply designing technology solutions that meet predefined business requirements. Instead, solution definition happens at the intersection of technological capability and business opportunity. Here, the crucial role of the Solution Architect is to ideate practical, technical possibilities that represent value to the business.

Participate in solution definition ☰ Defining solution **Capabilities** requires an understanding of the underlying architecture and technological strategies for implementation. For this reason, the Solution Architect is usually very involved in the solution definition as captured in **Solution Intent**. In addition to offering insights into the technical aspects of the solution, this involvement allows the architect to plan the necessary architectural **Enablers** to support the business value.

Evaluate technology tradeoffs ☰ The choice of technology for large solution development has significant implications. The Solution Architect assesses various possible technology alternatives and implementation approaches. In close collaboration with Solution Management, the Solution Architect makes strategic technology decisions, seeking a productive balance between the implementation cost and business value. Experimentation often informs such decisions.

Formulate architectural intent for the solution ☰ Solution Architects are responsible for defining the architectural concept of the solution. This includes critical technologies and their use, the structure of the solution in terms of its subsystems, and so on. This is one of the core aspects of the Solution Architect's role. The architect must define the architectural direction while fully understanding that the actual system design will evolve due to the implementation and that the original architectural ideas will undergo multiple adjustments. It is the task of the

Solution Architect not only to define architectures but also to see them validated in practice.

Define compliance requirements and solution NFRs ☰ The Solution Architect helps define the [Nonfunctional Requirements](#) (NFRs) for the solution based on the business need and the necessary technology tradeoffs. In addition, large solutions are often subject to regulatory compliance and standards, whose architectural and technological impact is managed by the Solution Architect.

Understand and communicate Solution Context ☰ Customer solutions live in an ecosystem that includes the execution environment, adjacent solutions, and other vital parameters. Collectively, this is known as [Solution Context](#). The solution context informs architectural decisions for the solution, its supporting infrastructure, and lifecycle processes. The Solution Architect communicates the solution context to the solution train and works with ARTs, suppliers, and shared services to design and implement a viable solution.

Defining and Communicating Architecture Vision

Solution architecture is constantly evolving to support the ongoing flow of business capabilities. The architectural changes need to be defined, their impact must be assessed, and all of this must be communicated to the many solution builders involved in the development effort. The Solution Architect achieves this via the following activities:

Prepare and update the architecture vision and roadmap ☰ To inform the direction of the Solution Train in the PI, the Solution Architect regularly updates the vision and strategy for solution architecture. The architect also continually updates the solution Roadmap with the architectural capabilities that guide the progression of the Solution Train over time.

Define architectural enablers and runway ☰ To support the Solution Train's business priorities, the System Architect defines the [Architectural Runway](#), comprised of enabler capabilities. It's a rolling-wave process that creates reasonable certainty, clarity, and detail for near-term needs while supporting flexibility for the future.

Participate in Pre- and PI Planning ☰ In [Pre-Planning](#) and [PI Planning](#) events, the Solution Architect communicates the architecture vision and works with System Architects and Suppliers to transform the high-level architecture intent into a realistic, actionable set of steps factored into the Solution Train's PI plan. While pre-planning occurs at the Solution Train level, the Solution Architect participates in the PI planning events with the constituent ARTs, as needed.

Ensure capacity allocation for enablement work ☰ Evolving and refactoring architectural capabilities require team implementation effort. The Solution Architect works with the Solution Management to define a productive capacity allocated for business capabilities vs. capacity for technology enablement.

Evolving Solution Architecture with ARTs and Suppliers

With Agile, the evolution of the solution's architecture does not happen in a top-down manner. Instead, the Solution Architect ensures that the high-level architectural intent evolves hand-in-hand with the hands-on implementation by the Agile Release Trains and suppliers. The following is done to support this process:

Support exploration and experimentation ☰ The Architectural runway contains numerous unknowns that need to be resolved to enable the intended business capabilities. The Solution Architect participates in the most critical explorations and experimentation aiming to address such unknowns. They often work with specific ARTs, suppliers, and individual [Agile Teams](#) in technical experimentation.

Apply modeling and simulation ☰ To approximate the critical aspects of the solution design in a timely and cost-efficient way, modeling and simulation are used—guided by the Solution Architect and implemented by the ARTs and suppliers. Solution Architects apply [Model-Based Systems Engineering](#) (MBSE) and simulation to progressively uncover and address the intricacies of near-term architectural advancements.

Regularly review solution increments ☰ The most convincing evidence of successful architecture is the functionality of the solution it enables. To this end, the Solution Architect regularly participates in [System](#) and [Solution Demos](#) and ensures that the architectural enablers achieve their purpose. SAFe recommends that a fully (or at least partially) integrated solution increment is produced multiple times in the PI (see [Enterprise Solution Delivery](#) for more detail on frequent integration).

Collaborate with ARTs and suppliers on solution design ☰ To build a viable solution, architectural intent needs to be balanced against the reality of implementation. To achieve this, the Solution Architect regularly interacts with System Architects and technology leads from ARTs and suppliers to collaborate on solution design and familiarize ARTs and suppliers with an end-to-end view of the solution.

Participate in establishing Solution Train flow ☰ Solution architecture is intricately connected with the structure of the development organization, each influencing the flow of customer value. The Solution Architect participates in establishing [Solution Train Flow](#) via the value stream mapping process and helps the Solution Train define improvements to the value delivery process. In this process, the architect often utilizes the “Inverse Conway Maneuver” by helping teams apply [Team Topologies](#).

Fostering Built-in Quality and the Continuous Delivery Pipeline

With large solutions, the impact of quality issues mounts up very quickly. Additionally, if done infrequently, solution

delivery creates substantial disruption and rework for the Solution Train. The Solution Architect is well-positioned to assist the train in establishing effective quality practices and building the continuous delivery capability through the following activities:

Foster solution design that supports frequent integration and testability ☰

Architectural choices influence how easy or hard it is to leverage [Built-in Quality](#) practices. The solution NFRs may further constrain the choices. Assisted by System Architects and Suppliers, Solution Architects help navigate the tradeoffs that enable frequent integration and testability of solution assets. Additionally, the Solution Architect is well-positioned to suggest specific implementation paths that leverage incremental development by supporting frequent integration and improved testability.

Participate in the design of the CDP ☰ The architect supports the creation and evolution of the [Continuous Delivery Pipeline](#). They typically provide guidance regarding technologies and tools to power the CDP. In addition, the Solution Architect directs solution design towards improved deployability by reducing and mitigating coupling across solution components and improving interfaces with other systems.

Define and evolve development and deployment infrastructure ☰ To ensure built-in quality practices are applied effectively and enable the CDP, the architect helps understand and define the infrastructure required to develop and deliver customer value. This work is often performed in close collaboration with the System Architects, Suppliers, System Team, and Shared Services responsible for deployment operations. Due to the complexity of the solution context, this may be a task with a high impact on the productivity of the development process.

Participate in release governance ☰ Based on their deep understanding of solution architecture and technology impact, the Solution Architect provides essential guidance regarding release governance. Their effort often involves determining the technology impact of specific releases and infrastructure implications. In addition, based on their understanding of the Solution Roadmap, the Solution Architect helps architecture evolve to support low coupling and improved deployability of subsystems and components where the change is most frequent.

Evolving Live Solutions

Once the initial version of the solution has been delivered to the customer, the development work usually doesn't end. Additionally, the solution context is often evolving, driven by increasing usage characteristics, etc. All of this needs continuous architectural effort in supporting the

usage stage of the solution and requires the following from the Solution Architect:

Evolve architecture to support new business priorities Ⓛ The architectural work does not stop after the initial version of the solution has been developed and released. New business capabilities will be planned, requiring architectural enablement to continue to be built into the solution progressively. To address this, the architect continuously updates and refines the architectural runway.

Advance and support NFRs Ⓛ As the solution goes live, matures, and undergoes various usage milestones, the landscape of applicable NFRs changes. The Solution Architect defines and communicates changes to existing NFRs and defines new NFRs needed to support evolving uses of the system.

Ensure continuous compliance Ⓛ The Solution Architect ensures that the new releases of the live solution adhere to regulatory requirements. This work may involve incorporating new compliance regulations. Additionally, the ones supported in the past are often affected by ongoing development and may require further work to ascertain compliance. Approaches include incorporating enablement for some compliance requirements into the solution design and using automated methods for compliance verification.

Last Update: 12 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

[Download SAFe
Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The epiphany of integration points is that they control product development and are the leverage points to improve the system. When timing of integration points slips, the project is in trouble.

ÑDantar P.
Oosterwal

Adjust Cookie Settings

Principle #4 → Build incrementally with fast, integrated learning cycles

In traditional, phase-gated development, investment costs begin immediately and accumulate until a Solution is delivered. Often, little to no actual value is provided before all of the committed Features are available, or the program runs out of time or money. During development, getting any meaningful feedback is challenging because the process isn't designed for it. Moreover, the development process itself isn't set up or implemented to allow incremental capabilities to be evaluated by the customer. As a result, the risk remains in the program until the deadline and even into deployment and after initial use.

No wonder the process is error-prone and problematic, often resulting in a loss of trust with the customer. Attempting to adjust for this, both parties try even harder to define the requirements and select the best design up-front. They also typically implement even more rigorous phase gates. Each of these remedies, unfortunately, actually compounds the underlying problem. This is a systems-level problem in the development process: it must be addressed systemically.

Faster Learning through Faster Cycles

The better approach begins with establishing a systematic way to accelerate learning. As originally defined by Shewhart and advanced by Deming [3], the plan-do-check-adjust (PDCA) cycle is the mechanism for gaining faster feedback and controlling the variability of solution development [3]. In many aspects, PDCA is simply an expression of the scientific method for creating a hypothesis, experimenting, and evaluating the results to navigate uncertainty and create new learning. It is an iterative cycle, as illustrated in Figure 1. Regardless of the results, every experiment creates new insights that help form the following hypothesis for learning to be gained or uncertainty to be resolved.

Figure 1. PDCA cycles

Figure 1. PDCA cycles

Fast feedback accelerates the acquisition of knowledge. Shorter PDCA cycles improve learning efficiency by decreasing the time between action and effect. They reduce the cost of risk-taking by truncating unsuccessful paths quickly. In other words, the shorter the cycles, the faster the learning. Smaller batch sizes and modern, automated development environments are two key moves that enable shorter cycles.

Integration Points Control Product Development

Establishing shorter PDCA cycles enables more frequent integration points. In complex systems development, local integration points are used to ensure that each system element or capability meets its responsibilities to contribute to the solution Intent. These local points must then be integrated at the next higher system level, as shown in Figure 2.

Figure 2. Synchronization of PDCA cycles

Figure 2. Synchronization of PDCA cycles

These integration points accelerate the learning process. However, development cannot proceed any faster than the slowest learning loop. Synchronizing the learning loops enables faster and more frequent integrations.

The larger the system, the more such integration levels exist. Solution designers recognize that the top-level, least-frequent integration point provides the only true measure of system progress and work to create these points as frequently as possible. All stakeholders understand that when the timing of integration points slips, the project is in trouble. But even then, this knowledge helps spark the necessary adjustments to scope, technical approach, cost, or delivery timing needed to redirect the effort to meet revised expectations.

Integration Points Reduce Risk

Any lack of integration points creates a false sense of security. This results in the belief that linear progress is being made when the problems are hidden in reality, only to be discovered later. This creates what Oosterwall [1] terms as “false positive feasibility,” i.e. the belief that the solution will be delivered as planned,

whereas, in reality, that is not the case, as Figure 3 illustrates.

Figure 3. The lack of frequent integration points creates false positive feasibility.

Figure 3. The lack of frequent integration points creates false positive feasibility.

Lean principles and practices approach the problem differently. Rather than pick a single requirements-and-design choice early on—assuming that it's feasible and will provide fitness for purpose—a range of requirements and design options ([Principle #3 D Assume variability and preserve options](#)) are considered while building the solution incrementally fast, integrated cycles. Each results in an increment of a working system that can be integrated and evaluated. Subsequent cycles build on the previous increments, and the solution evolves until it's released. The knowledge gained from integration points is not solely to establish technical viability. Many integration points reduce risk by providing minimum viable solutions or prototypes for testing the market, validating usability, and gaining objective customer feedback. Where necessary, these fast feedback points allow teams to pivot to an alternate course of action that should better serve the needs of the intended customers.

Integration Points Occur by Intent

The development process and the solution architecture are designed, in part, to focus on cadence-based integration points. Each point creates a “pull event” that pulls the various solution elements into an integrated whole, even though it addresses only a portion of the system intent. Integration points pull the stakeholders together, creating a routine synchronization that helps assure that the evolving solution addresses the real and current business needs instead of the assumptions established at the beginning. Each integration point delivers its value by converting uncertainty into knowledge:

Knowledge of the technical feasibility of the current design choice

Knowledge of the potential sustainability of the solution based on objective measures

(Principle #5 → Base milestones on objective evaluation of working systems)

Learn More

[1] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. Amacom, 2010.

[2] Ward, Allan C., and Durward Sobek. *Lean Product and Process Development*. Lean Enterprise Institute Inc., 2014.

[3] Deming, W. Edwards. *Out of the Crisis*. MIT Press, 2000.

Last update: 28 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The more alignment you have, the more autonomy you can grant. The one enables the other.

—Stephen Bungay,
author and strategy
consultant

Agile Release Train

The Agile Release Train (ART) is a long-lived team of Agile teams that incrementally develops, delivers, and often operates one or more solutions in a value stream.

[Adjust Cookie Settings](#)

Details

ARTs are teams of [Agile Teams](#) that align to a shared business and technology mission. Each is a virtual organization (typically 50 – 125 people) that plans, commits, develops, and deploys together. ARTs are organized around the enterprise’s significant [Development Value Streams](#) and exist solely to realize the promise of that value by building and delivering [Solutions](#) that benefit the [Customer](#).

ARTs are cross-functional and have all the capabilities needed to define, build, validate, release, and, where applicable, operate solutions.

These capabilities allow the ART to deliver a *continuous flow of value*, as shown in Figure 1.

Figure 1. A long-lived Agile Release Train

Figure 1. A long-lived Agile Release Train

ART Characteristics

Organized Around Value

As virtual organizations, ARTs have all the people needed to define, deliver, and operate the solution, eliminating the functional silos that may exist, as shown in Figure 2.

Figure 2. Traditional functional organization

Figure 2. Traditional functional organization

In a “functional” organization, developers work with developers; testers collaborate with other testers; architects and systems engineers work with each other, and operations work by themselves. Although there are reasons why organizations have evolved this way, the structure slows the flow of value, as it must cross all the silos. The daily involvement of managers is necessary to move the work across silos. As a result, progress is slow, and handoffs and delays rule the day.

Instead, the ART applies systems thinking (SAFe Principle #2) and organizes around value (SAFe Principle #10) to build an optimized cross-functional organization. This

facilitates the flow of value from ideation through deployment and release into operations, as Figure 3 illustrates.

Figure 3. Agile Release Trains are fully cross-functional

Figure 3. Agile Release Trains are fully cross-functional

Together, this fully cross-functional organization—whether physical (direct organizational reporting) or virtual (line of reporting is unchanged)—has everyone and everything it needs to define, deliver, and operate solutions. It is self-organizing and self-managing. This creates a far leaner organization, where traditional daily task and project management is no longer required. Value flows more quickly with less overhead.

To simplify the job of finding the optimum structure of ARTs within the organization and Agile teams within ARTs, SAFe recommends team topologies as described in the book *Team Topologies* [1]. SAFe recommends four ways to organize teams (Figure 4).

Stream-aligned teams are end customer-aligned and are capable of performing all the steps needed to build end-to-end customer value.

Complicated subsystem teams are organized around critical solution subsystems.

They focus on areas of high technical specialization, which limits the cognitive load on all the teams.

Platform teams provide application services and APIs for stream-aligned teams to be able to leverage common platform services.

Enabling teams provides tools, services, and short-term expertise to other teams.

Figure 4. Applying team topologies to Agile teams on an ART

Figure 4. Applying team topologies to Agile teams on an ART

Further guidance on organizing Agile teams can be found in the extended guidance article [Organizing Agile Teams and ARTs: Team Topologies at Scale](#).

Agile Teams Power the Train

ARTs include the Agile teams that define, build, and test features, as well as those that deploy, release, and operate the solution. SAFe Agile teams apply [SAFe Scrum](#) or [SAFe Team Kanban](#) or hybrids that suit their specific context. Each Agile team typically has ten or fewer dedicated individual contributors covering all the roles necessary to build a quality increment of value. Teams may be technology-focused

Ñdelivering software, hardware, and any combinationÑbusiness-focused, or a combination of both. Each Agile team has two specialty roles, the [Scrum Master / Team Coach](#) and the [Product Owner](#). And, of course, Agile teams within the ART are themselves cross-functional, as shown in Figure 5.

Figure 5. Agile teams are cross-functional

Figure 5. Agile teams are cross-functional

Aligned to a Common Cadence

ARTs also address one of the most common problems with traditional Agile development: teams working on the same solution operate independently and asynchronously. That makes it extremely difficult to routinely integrate the entire system. In other words, ÔThe teams are iterating, but the system isnÕt.Õ This increases the risk of late discovery of issues and problems, as shown in Figure 6.

Figure 6. Asynchronous Agile development

Figure 6. Asynchronous Agile development

Instead, the ART applies cadence and synchronization to assure that the system is iterating as a whole (Figure 7).

Figure 7. Aligned development: this system is iterating

Figure 7. Aligned development: this system is iterating

Cadence and synchronization ensure the focus remains on the evolution and objective assessment of the full system rather than its elements. The [System Demo](#), which occurs at the end of every [Iteration](#), provides objective evidence that the system is iterating. As Figure 7 illustrates, a [System Team](#) is often formed as an enabling group to help with infrastructure development and full system integration and validation. Over time, however, many of the centralized services provided by the System Team can be automated or absorbed by the teams themselves.

Enabled by Critical Roles

In addition to the Agile teams, the following roles aid the successful execution of the ART:

[Release Train Engineer \(RTE\)](#) is a servant leader who facilitates ART execution, impediment removal, risk and dependency management, and continuous improvement.

Product Management is largely responsible for what gets built, as defined by the **Vision**, **Roadmap**, and new **Features** in the **ART Backlog**. They work with customers, teams, and Product Owners to understand and communicate their needs and participate in solution validation.

System Architect is an individual or team that defines the system's overall architecture. They work at a level of abstraction above the teams and components and typically define **Nonfunctional Requirements (NFRs)**, major system elements, subsystems, and interfaces.

Business Owners are key stakeholders of the ART, with final responsibility for the business outcomes of the train.

Customers are the ultimate economic buyers or value users of the solution.

In addition to these critical ART roles, the following functions play an essential part in ART success:

System Teams typically assist in building and maintaining development, continuous integration, and test environments.

Shared Services are specialists necessary for the success of an ART but cannot be dedicated to a specific train. They often include data security, information architects, site reliability engineering (SRE), database administrators (DBAs), and many more.

Key ART roles rely on support from the teams. For example, the RTE depends on Scrum Master / Team Coaches for help with aspects of ART operations and improvement. Product Management relies on Product Owners to turn their product vision into reality. Architects collaborate with technology professionals on teams to devise viable architectures.

ART Responsibilities

The ultimate purpose of every ART is to deliver effective solutions to the customer. Essentially, ARTs are built for the sole purpose of establishing a fast flow of solution features. To achieve that, a train develops the solution iteratively, constantly engaging with the customer and adjusting the course of action towards an optimal solution.

Figure 8 shows the critical areas of responsibility of an ART that help achieve that objective:

Figure 8. ART responsibilities

Figure 8. ART responsibilities

Connecting with the Customer

Customers are the ultimate beneficiaries of the business solutions ARTs create and maintain. But connecting with the customer requires deliberate effort and a clear understanding of how to apply lean and agile practices in a unique ART context.

Apply customer centricity ▷ An ART routinely focuses on customer needs and opportunities to benefit the customer. **Customer Centricity** is a necessary mindset for the ART and its constituent teams. The ART works to increase and maintain customer empathy and continuously research better ways to solve customer problems.

Use design thinking ▷ A recurrent process of understanding the problem and designing the right solution. **Design Thinking** enables an ART to create desirable, feasible, and sustainable solutions. Paying close attention to user personas, journey mapping, and customer benefit analysis helps an ART discover new, valuable product capabilities. The use of lightweight prototypes validates customer value hypotheses quickly and keeps the ART on the right track.

Planning the Work

Planning crucial activities for an ART enables alignment across teams and stakeholders in terms of what and how to build within the next timebox. Alignment is one of the [Core Values](#) of SAFe, and ARTs, as a building block of a SAFe organization, have built-in means for achieving and sustaining alignment.

Align ART priorities with portfolio strategy

Every ART operates in a broader portfolio context and needs to align with the overall portfolio strategy. [Strategic Themes](#) orient the ARTs within a portfolio towards a common goal. However, achieving the alignment also requires an established process that involves: 1) regularly engaging with portfolio stakeholders at the ART level and 2) including ART representatives in portfolio interactions. Organizing this communication and interaction is easier around PI cadence. [Epic Owners](#) often serve as an important link between portfolio strategy and ART execution.

Prepare for PI Planning Stakeholders and teams need to prepare carefully for [PI Planning](#). Product Management and Business Owners develop the vision and agree on priorities for the next PI: teams take inventory of their remaining work, their attainable capacity, and any new effort that may emerge in the local context.

Plan the PI PI planning generates alignment within the ART. Teams create and agree on the [PI Objectives](#) that will guide them throughout the PI execution. Business Owners have an opportunity to share business and customer context with teams and, in turn, learn how the current technology and delivery capability can be employed to create optimal business value for the enterprise.

Delivering Value

ARTs develop solution features by applying a cadence that involves key activities to keep the train on the tracks. At certain points, an ART will release the newly created value to the customer.

Frequently integrate and test

A fast development rhythm requires frequent integration and testing. This helps uncover technology and implementation problems early and gives the teams enough time to respond to the findings. Without recurring integration and testing, an ART operates in excessive uncertainty and variability. [Built-in Quality](#) and [Team and Technical Agility](#) provide guidance on these practices.

Develop in short increments of value ☰ An ART implements the PI as a series of short increments, each representing a small batch of integrated, tested, and demonstrable value. The ART’s iteration cadence provides a natural pace to create these increments. Each helps the ART learn about potential implementation challenges, get customer feedback, and agree on a decision point with possible course corrections for the rest of the PI.

Regularly synchronize and make adjustments ☰ While executing the PI, an ART has multiple checkpoints in the form of an ART Sync, which includes a Coaches Sync and PO Sync (see the [PI](#) article for further description). These events increase visibility into the progress toward the current PI objectives and help the ART make timely adjustments.

Build a continuous delivery pipeline ☰ An effective Agile development process provides the means for ongoing exploration and integration of work. Additionally, the teams need to establish a continuous deployment process via building a [Continuous Delivery Pipeline](#) (CDP). This requires value stream mapping to identify the sources of excessive delay and variability. As a part of CDP, [Continuous Deployment](#) often involves purposeful system design that favors low coupling of capabilities, which enables the teams to deploy value independent of each other.

Establish release governance process ☰ Each ART establishes a governance process suitable for its release cycle. The governance process includes the ways to plan and execute the releases. This involves several activities, including:

- Aligning releases with strategic goals
- Validating releasable increments
- Ensuring compliance with standards and regulations
- Assessing customer impact
- Maintaining the supporting assets and activities for releasing

Release frequently and continually optimize the process ☰ Releasing frequently helps reduce time-to-market. Additionally, establishing successful continuous delivery and governance processes is only possible when the releases happen on a frequent, reliable basis. Over time, solution assets, architectures, and the infrastructure evolve and accumulate technical debt that may unexpectedly disrupt the release process. Releasing regularly helps uncover, mitigate, or even prevent those issues before they cause damage.

Getting Feedback

Getting fast feedback is the primary component of an ART’s high development velocity: speed comes from fast learning and adaptation rather than from “working harder.” Technology feedback results from integration and testing as well as running technical spikes. The feedback on the product value comes from the customer and business stakeholders. ARTs routinely:

Involve the customer in the development process ☰ There is no substitute for direct customer input. Including it in a routine development process helps an ART move at a much higher speed to avoid the costly mistake of building capabilities the customer doesn't need or cannot use. Preparation for PI planning, the PI planning itself, and system demos provide venues for customer interaction.

Measure business outcomes and usage ☰ Customer use of solutions may reveal issues and opportunities that otherwise might remain invisible to the ART. Creating the data capture and analytics capabilities, however, requires investment in the train's capacity, a proactive approach, and the use of [Architectural Runway](#). Additionally, an ART must measure whether delivered solutions enable the desired business outcomes—the ultimate purpose of the ART's effort.

Perform routine A/B testing ☰ Successful solution development is contingent upon an ART's ability to navigate the unknowns and make effective decisions. A/B testing enables effective decision-making and improves the development speed of an ART. Instead of prematurely committing to certain functionality, the ART creates two or more options and validates them with users, thus gaining a real sense of which alternative is performing better.

Test User Experience ☰ User Experience (UX) is essential to fully realizing the solution potential. But to provide productive UX, there needs to be an explicit, thorough UX design and testing strategy. As a part of this process, hypotheses are formulated, and then Minimum Marketable Features (MMF) are built and evaluated by observing the user in action, surveying them, or utilizing analytics. The SAFe [Lean UX](#) article covers additional topics of enabling effective UX.

Improving Relentlessly

An ART seeks to continuously improve productivity in delivering customer value. Naturally, the process requires measuring different aspects of ART operations and identifying areas for improvement:

Measure competency, flow, and outcomes ☰ Every ART should regularly assess against key applicable competencies. ARTs also should routinely measure [ART Flow](#) and apply [Flow Accelerators](#) to initiate forward momentum for ongoing flow improvement. Additionally, ARTs use their [Value Stream KPIs](#) to measure the outcomes that underpin the desired customer and business benefits.

Inspect & Adapt at regular intervals ☰ At every PI boundary, an ART has an opportunity to look back at the last PI, identify problems, and take corrective action during the [Inspect & Adapt](#) (I&A) event. This is the perfect time to identify significant, systemic improvement opportunities.

Make small improvements on the fly ☰ Every ART routinely discovers small, local, and tactical improvement opportunities. In most cases, it is best to address these as they occur and without waiting for the next I&A. This achieves quick wins and preserves the I&A for issues that require more attention and the involvement of high-

profile stakeholders.

Leverage Innovation & Planning Iteration Ⓜ The IP Iteration offers an opportunity to allocate uninterrupted time to innovation and learning. This helps the ART to further advance its solution, technical infrastructure, and various processes.

Learn More

[1] Skelton, Mathew, and Manuel Pais. *Team Topologies*. IT Revolution Press, 2019.

Last update: 24 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd, Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



*The emphasis
should be on why
we do a job.*

ÑW. Edwards Deming

ART and Solution Train Backlogs

The ART Backlog is a Kanban system that is used to capture and manage the features and enablers intended to enhance the solution and extend its architectural runway.

Adjust Cookies Settings The Solution Train Backlog is a Kanban system that is used to capture and manage the capabilities and enablers intended to enhance the large solution and extend its

architectural runway.

The ART and Solution backlogs capture the Solution's upcoming [Features](#), Capabilities, and [Nonfunctional Requirements \(NFRs\)](#). Managing the backlog is a critical economic driver for trains and the portfolio. [Product Management](#) has responsibility for the ART Backlog, while [Solution Management](#) is responsible for the Solution Train Backlog. These backlogs are visualized and managed in Kanban systems, where features, capabilities, and enablers are captured, defined, evolved, and prioritized to ensure a continuous flow of value to [Customers](#).

Details

Product and Solution Management develop, maintain, and prioritize the ART and Solution Train backlogs. They actively collaborate with stakeholders, including Customers, [Business Owners](#), [Product Owners](#) (POs), [System](#) and [Solution Architects](#), and others (for example, RTEs/STEs) to discover the features and capabilities needed to advance the solution.

Figure 1 illustrates a view of the ART and Solution Train backlogs with their primary input sources:

- 1) [Portfolio Epics](#) split into features or capabilities
- 2) Capabilities and enablers arising from the Solution Train's local context
- 3) Solution Train capabilities split into features and enablers
- 4) Features and enablers created from the ART's local context

Figure 1. Input sources for the ART and Solution Train Backlogs

Figure 1. Input sources for the ART and Solution Train Backlogs

An effective solution must meet all its functional and Nonfunctional Requirements (NFRs). NFRs shown at the bottom of the backlog in Figure 1 serve as constraints or restrictions across the backlog, affecting the solution's design and performance. They are typically captured in *acceptance criteria* or as part of the definition of done (DoD). NFRs are persistent qualities, and ARTs often revisit them as part of the DoD.

Building and Refining the Backlog

Product and Solution Management take a continuous, flow-based approach to refine their backlogs, ensuring features and capabilities are ready for implementation within the appropriate level of discovery and risk. Refining the ART and Solution Train backlogs to ensure readiness often involves the following activities:

- Discovering new features and capabilities
- Reviewing and updating backlog item definitions, including developing acceptance criteria and benefit hypothesis
- Identifying the enablers required to support new features and capabilities
- Applying [Behavior-Driven Development](#) (BDD) techniques to help clarify features and capabilities or holding specification workshops
- Prioritizing the backlogs using [Weighted Shortest Job First \(WSJF\)](#) in collaboration with Business Owners, System Architects, POs, and other stakeholders, such as RTEs/STEs
- Briefing Agile Teams and stakeholders about upcoming features and capabilities for [PI Planning](#)
- Deleting aging and no longer relevant items

Refinement activities often occur during the PO Sync, where Product Management and Product Owners identify new backlog items, revise others, and remove obsolete items. A well-maintained backlog is a prerequisite for a successful PI planning event and execution.

Managing the Backlog with Kanban

The ART and Solution Train Kanban system facilitates the flow of [Features](#) and Capabilities through the [Continuous Delivery Pipeline](#). Figure 2 illustrates a *typical* ART Kanban with *example* policies and WIP limits governing each state. While this is a good starting point, the system should be adapted to fit the train's needs, including the definition of WIP limits and the specific policies for each state.

Figure 2. A typical ART or Solution Train Kanban system

Figure 2. A typical ART or Solution Train Kanban system

The following process states describe the example ART and Solution Train Kanban flow.

Funnel -All new big ideas are welcome here, typically expressed as Features or Capabilities. These ideas may indicate needed new functionality, enhancement of the existing system functions, or [Enablers](#).

Analyzing ☰ New ideas that align with the Solution [Vision](#) and [Strategic Themes](#) are further explored by [Agile Teams](#) when they have available capacity. Analyzing includes Continuous Exploration activities (e.g., Customer Centricity, Design thinking) and collaboration to create one or more well-formed features (see backlog refinement above). The WIP limit for this state considers the availability of Product and Solution Management, the capacity of teams, and other subject matter experts.

Ready ☰ The highest-priority features analyzed and approved by Product Management or Solution Management advance to this state. They are prioritized

with [WSJF](#) relative to the rest of the backlog and await implementation.

Implementing Features are pulled into the Implementing state as teams start working on them. Teams implement them throughout the PI.

Validating On Staging The *validating on staging* step has two sub-states, *In Progress* and *Ready*:

In Progress Features that are implemented and ready for feedback get pulled into this state. Teams integrate and test them with the rest of the system in a staging environment and present them for approval.

Ready Approved features move to this state's "Ready" buffer, where they are prioritized again using WSJF to await deployment.

Deploying To Production The *deploying to production* step also has two sub-states, *In Progress* and *Ready*:

In Progress Features get moved to production immediately in an automated Continuous Delivery environment or to the *in-progress* state for manual deployment when capacity exists.

Ready ARTs that separate deployment from release move the item to the *ready* buffer of *deploying to production* to await release. In other cases, features are automatically released, and users can immediately access them. This state is WIP limited to avoid the buildup of deployed but not yet released ñ features.

Releasing When there's sufficient value, market needs, and opportunity, features are released to some or all customers to evaluate their benefit hypothesis.

Done After a feature has been released and evaluated, it moves to the *done* state. However, new work items may be created based on customer feedback.

Managing ART and Solution Train Epics

Some ART and Solution Train initiatives are too big to be completed in a single PI. These ART or Solution Train [Epics](#) are typically identified and managed in a separate Kanban system, as shown in Figure 3. Also, some portfolio epics may require splitting into ART and Solution Train epics to facilitate incremental implementation. While mainly a local concern, these epics may impact financial, human, and other resources that might be large enough to warrant a Lean business case, discussion, and financial approval from [Lean Portfolio Management \(LPM\)](#). Epics that exceed the portfolio epic threshold require review and approval. This is a critical [Guardrail](#) on budgetary spending.

The primary purpose of this Kanban system is to analyze and approve ART and Solution Train epics, splitting them into features or capabilities that will be further explored and implemented using the ART or Solution Train Kanban. These Kanban systems may not be required depending on how frequently ART and Solution Train epics occur in the local context.

Figure 3. A typical ART or Solution Train Epic Kanban

Figure 3. A typical ART or Solution Train Epic Kanban

The process states in the ART or Solution Train Epic Kanban are like the [Portfolio Kanban](#), for example:

Funnel ☰ All big initiatives are welcome in the “Funnel” state. There is no WIP limit.

Reviewing ☰ Subject Matter experts (SMEs) and stakeholders review the epics and prioritize them using [WSJF](#) and other criteria to determine which ones should move on for more in-depth exploration. If the item exceeds the portfolio epic threshold established by LPM, they move to the Portfolio Kanban. The WIP limit for this state considers the availability of SMEs.

Analyzing During this exploration state, SMEs and stakeholders do the following types of analysis:

- Refine size estimates, and WSJF relative to other epics
- Consider solution alternatives
- Identify possible Minimum Marketable Features (MMF) or Minimum Viable Products (MVPs)
- Forecast the costs, and identify technology, architectural enablement, and infrastructure using a Lean business case (described in the epics article)

Guided by analysis and insights, Product and Solution Management and Business Owners (and LPM where required) approve or reject the epics. Approved epics are split into features or capabilities and moved to the ART or Solution Train Kanban funnel, where they will be prioritized based on WSJF. WIP limits apply to the analyzing state.

Like the portfolio Kanban, ART and Solution Train epics typically require [Epic Owners](#) to help define, explore, and implement them.

Balancing Value Delivery and Solution Health with Capacity Allocation

Every train needs to balance the backlog of business features or capabilities with continuous investment in enablers to build and maintain the architectural runway, avoiding velocity reduction and technology obsolescence. Further, these enablers support exploring requirements and applying design thinking for future PIs, creating prototypes and models, and enhancing visibility into opportunities and problem areas.

The collaboration during WSJF prioritization is often sufficient to communicate concerns and arrive at a good balance of work. When that approach is insufficient, Product and Solution Management may work with Architects to apply capacity allocation to decide how much of the total effort the ART will reserve for each activity type for the upcoming PI. Figure 4 illustrates an example of capacity allocation.

Figure 4. Example capacity allocation for a PI

Figure 4. Example capacity allocation for a PI

While the agreed-to capacity allocation can persist for several PIs, it should be periodically reviewed and adjusted during backlog refinement in preparation for PI planning.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[3] Reinertsen, Don. *Principles of Product Development Flow: Second Generation*

Lean Product Development. Celeritas Publishing, 2009.

Last update: 14 March 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)

- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



*Our highest priority
is to satisfy the
customer through
early and
continuous delivery
of valuable software.*

~Agile Manifesto

Continuous Delivery Pipeline

The Continuous Delivery Pipeline (CDP) represents the workflows, activities, and automation needed to guide new functionality from ideation to an on-demand release of value.

Figure 1 illustrates the pipeline's four aspects: [Continuous Exploration \(CE\)](#), [Continuous Integration \(CI\)](#), [Continuous Deployment \(CD\)](#), and [Release on Demand](#).

Figure 1. The SAFe Continuous Delivery Pipeline

Figure 1. The SAFe Continuous Delivery Pipeline

The Continuous Delivery Pipeline (CDP) is a significant element of the [Agile Product Delivery](#) competency. Each [Agile Release Train \(ART\)](#) builds and maintains, or shares, a pipeline with the assets and technologies needed to deliver solution value as independently as possible. The first three elements of the CDP (CE, CI, and CD) work together to support the delivery of small batches of new functionality, which are then released to fulfill market demand.

Details

Building and maintaining a CDP allows each ART to deliver new functionality to

users far more frequently than traditional processes. For some, continuous may mean daily or even releasing multiple times per day. For others, it may mean weekly or monthly releases—whatever satisfies market demands and the goals of the enterprise.

Legacy practices often cause ARTs to make solution changes in large monolithic chunks. However, this does not usually need to be an all-or-nothing approach. For example, a satellite system comprises a manufactured orbital object, a terrestrial station, and a web farm that feeds the acquired data to end users. Some components may be released daily—perhaps the web farm functionality or satellite software. Other elements, like the hardware components, can only be done once every launch cycle.

Decoupling the web farm functionality from the physical launch eliminates the need for a monolithic release. It also increases Business Agility by allowing the delivery of solution components in response to frequent market changes.

The Four Aspects of the Continuous Delivery Pipeline

The SAFe CDP contains four aspects: continuous exploration, continuous integration, continuous deployment, and release on demand. The CDP enables organizations to map their current pipeline into a new structure and use relentless improvement to deliver value to customers. Internal feedback loops often identify process improvements, while external feedback identifies solution improvements. The improvements collectively create synergy in ensuring the enterprise is “building the right thing, the right way” and frequently delivering value to the market. The paragraphs below describe each aspect.

Continuous Exploration (CE) focuses on creating alignment on what needs to be built. In CE, design thinking ensures the enterprise understands the market problem or customer need and the solution required to meet that need. It starts with a hypothesis of something that will provide value to customers. Ideas are then analyzed and further researched, leading to the understanding and convergence of the requirements for a Minimum Viable Product (MVP) or Minimum Marketable Feature (MMF). These feed the solution space for exploring how existing architectures and solutions can or should be modified. Finally, convergence occurs by understanding which **Capabilities** and **Features**, if implemented, are likely to meet customer and market needs. Collectively, these are defined and prioritized in the ART Backlog.

Continuous Integration (CI) focuses on taking features from the ART backlog and implementing them. In CI, the application of design thinking tools in the problem

space focuses on the refinement of features (for example, designing a user story map), which may motivate more research and the use of solution space tools (such as user feedback on a paper prototype). After specific features are clearly understood, Agile Teams implement them. Completed work is committed to version control, built and integrated, and tested end-to-end before being validated in a staging environment.

Continuous Deployment (CD) takes the changes from the staging environment and deploys them to production. At that point, they're verified and monitored to ensure they are working correctly. This step makes the features available in production, where the business determines the appropriate time to release them to customers. This aspect allows the organization to respond, rollback, or fix forward when necessary.

Release on Demand (RoD) is the ability to make value available to customers together or in a staggered fashion-based on market and business needs. This approach allows the business to release when market timing is optimal and carefully controls the risk associated with each release. Release on demand also encompasses critical pipeline activities that preserve solutions' stability and enduring value long after release.

Although described sequentially, the pipeline isn't strictly linear. Instead, it's a learning cycle that allows teams to establish one or more hypotheses, build a solution to test each one, and learn from that work, as Figure 2 illustrates.

Figure 2. The CDP fosters continuous learning and value delivery

Figure 2. The CDP fosters continuous learning and value delivery

Although a single feature flows through the Value Stream sequentially, the teams work through all aspects in parallel. That means that ARTs and Solution Trains, throughout every PI and every iteration in the PI, continuously:

- Explore user value
- Integrate and demo value
- Continuously deploy to production
- Release value whenever the business needs it

Start by Mapping the Current Workflow

Successful enterprises already have a delivery pipeline—otherwise, they wouldn't be able to release any value. But too often, they are not fully automated, contain significant delays, and require tedious and

Value Stream Mapping

error-prone manual intervention. These factors cause organizations to delay releases, increasing their size and scope (ØWeÕll release when it is big enough).

This approach is the opposite of the [SAFe Principle #6](#), which makes value flow without interruption.

The first step to improving value flow is mapping the current pipeline. Figure 3 illustrates the flow of value through one enterpriseÕs CDP, focusing initially on new Feature development. Over time, this map would be extended to capture any change to the system, from new Features to maintenance to architectural improvements.

Figure 3. A map of one company's delivery pipeline

Figure 3. A map of one companyÕs delivery pipeline

Once the current pipeline has been mapped, metrics can be collected and recorded on the value stream map to understand where delays occur. These metrics enable the ART to identify opportunities for improvement (such as eliminating delays or reducing rework). Four primary metrics [1] are used (Figure 4):

Process time is the time it takes to get work done in one step. For example, in Figure 4, the "Design" step takes four hours.

Lead time is the time it takes from when the work was done in the previous step until it's done in the current step. In other words, *lead time = delay time from the last step + process time* of the current step. In Figure 4, the lead time from creating ideas to defining them is variable. It is common when first mapping systems not to have metrics on specific steps. In this case, the ART can improve the remaining process while metrics are gathered on the variable part.

Delay time is the time when no work is happening. Figure 4 shows that the features accepted by the Product Manager are delayed a staggering 696 hours before being deployed to staging. Understanding and eliminating unnecessary delays is critical to improving the flow of value.

Percent complete and accurate (%C&A) represents the percentage of work that the next step can process without needing rework. Often, delays are caused by poor quality in the upstream (prior) steps. The %C&A metric helps identify the steps where poor quality might be occurring and causing longer lead times, resulting in delays in value delivery. Figure 4 indicates that 20% of the time, the work moving from the "Design" step to the "Code" step needs to be reworked. Improving the %C&A metric is also essential to improving the flow of value. The %C&A of a single step is extended to *rolled %C&A*, a measure that captures the likelihood that an item will pass through the *entire* workflow without rework. With a cumulative rolled %C&A of 35%, this workflow is reworking more than half of its steps.

Figure 4. Value stream map with flow metrics

Figure 4. Value stream map with flow metrics

Align the Current Workflow to the Continuous Delivery Pipeline

Once the current flow is understood, it can be mapped into the SAFe CDP. Mapping helps the organization adopt a shared mental model and efficiently communicate changes and improvements. Figure 5 removes the continuous labels because, at this stage, the process is unlikely to resemble an automated pipeline.

Figure 5. SAFe's Continuous Delivery Pipeline mapped to a value stream

Figure 5. SAFe's Continuous Delivery Pipeline mapped to a value stream

Identify Opportunities for Improvement

Teams look for the opportunity to improve the efficiency of each step, consequently reducing the total lead time. This improvement includes addressing process time and each step's quality (percent complete and accurate). The higher that number, the less rework is required and the faster the work moves through the system.

As shown in Figure 6, the delay time between steps is often the most significant initial factor. Delay time represents handoffs, waiting, and other non-value-added wastes. This process has two considerable delays and a substantial amount of rework in the first step of the deployment process. Reducing delays is typically the fastest and easiest way to lower the total lead time. Another high-priority area to improve is any step with low %C&A metrics, as reducing rework enables the ART to focus on creating value (for example, for a software solution, instead of fixing bugs, the team can focus on new features). Subsequent opportunities for improvement

focus on reducing the batch size and applying the DevOps practices identified in each of the specific articles describing the continuous delivery pipeline. The [Team Flow](#) and [ART Flow](#) articles provide further guidance on how to make value flow without interruption (Principle #6).

Figure 6. Value stream maps reveal major delivery bottlenecks

Figure 6. Value stream maps reveal major delivery bottlenecks

Tracking Continuous Delivery

Continuous delivery is an extensive process. Indeed, it may be the most vital capability of every ART and Solution Train. [Product Management](#) and its stakeholders should visualize and track ongoing work, even though a significant portion of the development is automated. The ART Kanban facilitates the flow of [Features](#) through the CDP. Work in Process (WIP) limits help improve throughput and identify and address bottlenecks. Figure 7 illustrates a typical ART Kanban, example policies, and WIP limits governing each state.

Figure 7. An example ART Kanban board

Figure 7. An example ART Kanban board

The Kanban systems consist of a series of states, each of which is summarized below:

Funnel ⚡ This is the capture state for all new features or enhancement of existing system features.

Analyzing ⚡ Features that best align with the vision are pulled into the analyzing step for further exploration. Here they're refined with critical attributes, including the business benefit hypothesis and acceptance criteria.

Ready ⚡ After analysis, higher-priority features move to the backlog, where they're ranked.

Implementing ⚡ At every PI boundary, top features from the ART backlog are pulled into the implementing stage, where they're developed and integrated into the system baseline.

Validating on staging ⚡ Features ready for feedback get pulled into this step to be integrated with the rest of the system in a staging environment and then tested and

validated.

Deploying to production ⚡ When capacity is available, features are deployed into the production environment, where they await release.

Releasing ⚡ Features are released once a sufficient amount of value has been created to meet market demands and the benefit hypothesis is evaluated.

Done ⚡ When the hypothesis has been satisfied, no further work on the feature is necessary, and it moves to the done column.

Enable the Continuous Delivery Pipeline with DevOps

Building, maintaining, and optimizing a continuous delivery pipeline requires specialized skills and tools throughout the entire value stream. Because this type of delivery system calls for the rapid delivery of complex solutions with very short learning loops and high degrees of cross-functional collaboration, DevOps methods are ideally suited to enable it. In other words, continuous delivery pipelines are best implemented with DevOps, as illustrated in Figure 8.

Figure 8. DevOps enables the CDP

Figure 8. DevOps enables the CDP

SAFe's [CALMR](#) approach to DevOps is a mindset that guides continuous value delivery by improving *Culture, Automation, Lean Flow, Measurement, and Recovery*. DevOps technical skills, practices, and tooling are grouped into *practice domains*, represented by the model's inner loops. The two outer loops represent the four aspects of the CDP, each of which has four activities.

Download the free DevOps Health Radar assessment [here](#). AgilityHealth also offers an [online version of this assessment](#).

Learn More

- [1] Martin, Karen. *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation*. McGraw-Hill Education, 2013.
- [2] Kim, Gene. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*. IT Revolution Press, 2013.
- [3] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

Last update: 5 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS

TRAINING

[YouTube](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



While DevOps needs particular champions, excellence can be achieved only if the entire organization is on board.[1]

ÑDevOps Institute

DevOps Practice Domains

Adjust Cookie Settings

Note: This is the **third article** in the **SAFe DevOps series**. It introduces

the technical skills and tools that power the Continuous Delivery Pipeline.

The next two articles can be accessed from the following links: [A CALMR](#)

[Approach to DevOps](#) and [DevOps home page](#).

Enabling continuous flow through the end-to-end value stream in today's enterprises can be complex. Practitioners across multiple teams and functions must collaborate effectively. Quality and security must be built in at every step. Solutions must be architected for rapid change. Governance and compliance controls must be streamlined and automated. A slew of metrics must be managed from all layers of the solutions stack. And the right automation must be in place to enable lightning-fast flow and on-demand processing.

Managing all this complexity, let alone optimizing it for continuous flow, requires specialized knowledge from everyone in the value stream who influences product delivery. Succeeding with [DevOps](#) requires applying knowledge at the right time, by the right people, and with the right tools.

SAFe categorizes this knowledge into DevOps *practice domains*, which comprise the inner rings of Figure 1 and together enable the [Continuous Delivery Pipeline](#). Each contains a set of specific practices and tools that perform critical DevOps functions in the value stream. CALMR, at the center of figure 1, is not a practice domain; instead, it is the mindset that guides DevOps behavior and decision-making throughout the entire value stream. For details, please see the [CALMR article](#).

Figure 1. DevOps practice domains enable the CDP

Figure 1. DevOps practice domains enable the CDP

Some domains, such as *Continuous Quality* and *Continuous Security*, span the entire CDP. This breadth indicates that there is always work in those areas. Others, such as *Deployment Pipeline* and *Infrastructure Management*, apply to smaller segments of the CDP. Work in those domains enables a subset of activities in the value stream. At any given time, knowledge from multiple disciplines is combined with supporting the current activity in the continuous delivery pipeline, as explained in the next section.

The outer ring of practice domains, composed of *Agile Planning and Design*, *Deployment Pipeline*, and *Continuous Monitoring*, represents a solution's path through the system. It is where cross-functional DevOps practices converge with the evolving solution and is the "conveyor belt" that moves the solution through its delivery life cycle.

Details

DevOps implies close collaboration across team and department boundaries, as illustrated in Figure 2. Skills across domains are combined to produce specific outcomes in the CDP. For example, ensuring features are developed to foster continuous delivery. ARTs must apply knowledge from six domains (in this case, *Deployment Pipeline, Configuration Management, Version Control, Continuous Security, Continuous Quality, and Value Stream Management*) to develop solutions with speed, quality, security, version control, and more.

Figure 2. The CDP pulls DevOps practices in from across domains

Figure 2. The CDP pulls DevOps practices in from across domains

Using this model, it is clear where knowledge must Ÿshift leftŸ and which individuals may need to collaborate at each delivery stage.

McKinsey states, ŸTransformation success requires thoughtful mastery of multiple domains and coordination of many moving parts.Ÿ [2] This DevOps model guides the management of this complexity to achieve transformation success. The result is a continuous delivery pipeline driven by a cross-functional DevOps culture that can deliver valuable digital solutions on demand.

Achieving that result begins with an understanding of the technical practices that power this new way of building. Following is an introduction to each DevOps practice domain and the specific technical practices and tools it encompasses.

Value Stream Management

Value stream management (VSM) is a business practice that focuses on increasing the flow of business value from customer request to customer delivery [3]. VSM provides lightweight, end-to-end governance of the continuous delivery pipeline and optimizes it for maximum value delivery instead of complete adherence to fixed delivery plans.

VSM includes specific practices such as mapping value streams, analyzing flow efficiency through the end-to-end delivery pipeline, and setting delivery speed, quality, and value targets. VSM also involves specialized software platforms that integrate with other tools throughout the pipeline to collect and surface real-time data about the health of the value stream.

Continuous Quality

DevOps practices that ensure high solution quality do not simply shift left; they extend to all activities in the value stream. ***Continuous Quality*** is a domain that ensures that quality is built in early in the pipeline and continually managed throughout a solution's life cycle.

Continuous Quality includes specific practices like hypothesis-driven development, behavior-driven design (BDD), test-driven development (TDD), A/B testing, and exploratory testing. Automation enhances the speed and accuracy of testing throughout the value stream.

Continuous Security

Like quality, security is everyone's responsibility. This domain spans the entire continuous delivery pipeline and ensures reasonably good information security practices—preventive and detective—are always active throughout the solution's lifecycle.

Continuous Security emphasizes practices such as security by design, threat modeling, security-as-code, and automation in vulnerability scanning, penetration testing, and intrusion detection.

Version Control

DevOps requires all solution assets to be under version control to achieve quantum leaps in pipeline acceleration and quality control. Application code; server, network, and firewall configurations; database scripts; even requirements and test scripts need to be stored in a common repository to ensure ARTs can build solutions and environments and deploy, repair, and decommission them on demand.

Version Control stresses minimizing the amount and duration of open branches and ensuring that the code base always remains in a deployable state. Automation in this category is directed toward optimizing branching and merging strategies, facilitating distributed version control, and managing file types of all kinds.

Configuration Management

Configuration management systems have two main components: ***version control and task/issue management***. Version control software keeps configurations in its repository. They often include application, database, infrastructure, and security configurations. Since these configurations collectively represent all design time and runtime aspects of a solution, they need careful management.

The related practices in this domain are expressed Ôas codeÕin infrastructure-as-code, security-as-code, and compliance-as-code, for instance. The ***Configuration Management*** practice domain represents an Ôeverything as codeÕ approach to defining solutions, their runtime environments, and the systems that pull them through the continuous delivery pipeline.

Infrastructure Management

Infrastructure Management is a domain that focuses on the maintenance of production systems, many of which may be defined as code. Whereas configuration management is a set of design-time practices, infrastructure management is a set of runtime practices.

This domain is primarily concerned with ensuring the stability and resiliency of deployed solutions so that the business can realize the maximum value. Critical technical practices include chaos engineering, selective releases, and site reliability engineering (SRE). Automation is applied to enable immutable infrastructure, release on demand, rapid recovery, and auto-scaling.

Agile Planning and Design

DevOps begins well before development work starts. This approach ensures that the *inputs* to development, such as desired business outcomes, solution scope, architecture, and design, are performed in a way that fosters continuous delivery.

The *Agile Planning and Design* domain assures that ARTs follow a CALMR approach to DevOps from the onset of the delivery life cycle. This domain includes practices such as Lean UX, Agile architecture, backlog management, and emergent design. Tools focus primarily on streamlining Agile practices, making work visible, and accelerating the productivity of individuals and teams.

Deployment Pipeline

Arguably the most popular area of concentration in DevOps is the *CI/CD pipeline*. This segment of the value stream begins with code committed to version control and ends with that code being successfully deployed in production. Because the goal is to deploy finished code to production as quickly and safely as possible, SAFe's term for this subset of the value stream is the *deployment pipeline*.

Ultimately, the entire deployment pipeline should be automated from end to end using a carefully selected toolchain. Doing so accelerates builds, integration, testing, break-fix, and deployment cycles. Often the only manual activities in a mature deployment pipeline are peer reviews and responding to system-generated pipeline events and notifications.

Continuous Monitoring

Continuous Monitoring involves capturing, logging, surfacing, and analyzing production events. These events describe both system behavior and end-user behavior and directly support the *measurement* and *recovery* aspects of CALMR.

This domain calls for significant harmony between the monitoring systems and those who interpret and act on the results. This set of practices is tailored to measure and maintain business value with high precision through full-stack telemetry, observability, proactive issue detection, visualization, AIOps, and analytics.

Agile Product Management

DevOps enables a continuous learning loop, which drives rapid innovation. *Agile Product Management* defines what needs to be learned and ensures what was

learned is captured and quantifiable.

Through customer centricity, hypothesis-driven development, design thinking, Lean Startup, and market research practices, this domain ensures that the continuous delivery pipeline is always calibrated to deliver specific, measurable business outcomes. Moreover, it ensures that the enterprise collects validated learning from every release and directs those learnings toward relentless improvement.

For more detail, please see the [Product Management](#) article.

Value Metrics

The ultimate goal of DevOps is to enable enterprises to deliver continuous value to their customers. The **Value Metrics** domain provides a method for defining and measuring value from the customer's perspective.

Value metrics are precisely tuned to how customers derive value from deployed solutions. Outcome-oriented measurements provide objective evidence that solutions are (or are not) performing well in the market. This domain contains techniques such as innovation accounting, objectives and key results (OKRs), key performance indicators (KPIs), and service level agreements (SLAs). These measures forecast desired outcomes while evaluating actual outcomes determines a solution's true value to the business.

More in the DevOps Series

Article 1: [DevOps Main Page](#)

Article 2: [A CALMR Approach to DevOps](#)

Article 3: SAFe's DevOps Practice Domains (this page)

Learn More

[1] Upskilling: Enterprise DevOps Skills Report 2019. <https://devopsinstitute.com/wp-content/uploads/2019/03/UpskillingReport-Final.pdf>

[2] Agile's Next Level: ABM AMRO's Hybrid Cloud-DevSecOps Transformation.

<https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/agiles-next-level-abn-amros-hybrid-cloud-devsecops-transformation#>

[3] The Rise of Value Stream Management (VSM).

<https://www.linkedin.com/pulse/rise-value-stream-management-vsm-mik-kersten/>

Last update: 4 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

[GET SOCIAL](#)
[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

Scaled Agile Framework



*Everything must be
made as simple as
possible. But not
simpler.*

ÑAlbert Einstein

Large Solution SAFe

Large Solution SAFe is for enterprises building large and complex solutions that do not require portfolio concerns.

Adjust Cooker settings
Large Solution SAFe (Figure 1) describes the additional roles, events, and artifacts needed to help those who build and evolve the largest and most sophisticated

software applications, networks, and cyber-physical systems and includes the following constructs:

The [Essential SAFe](#) configuration

An additional competency, [Enterprise Solution Delivery](#), describes how to apply Lean-Agile principles and practices to the specification, development, deployment, operation, and evolution of the world's largest and most sophisticated software applications, networks, and cyber-physical systems.

The large solution-level roles, artifacts, and events

The full spanning palette

A connection to the [Enterprise](#) or [Government](#) entity the solution supports

Figure 1. Large Solution SAFe

Figure 1. Large Solution SAFe

Details

Large Solution SAFe builds on Lean systems engineering practices and artifacts to

provide a more flexible approach to the development, deployment, and operation of large solutions. It supports enterprises building and operating large-scale solutions that require significant collaboration beyond the scope of a single ART. Building these solutions requires additional roles, artifacts, events, and coordination.

The remainder of this article describes the highlights, roles, events, and artifacts at the large solution level (Figure 2).

Figure 2. Large Solution Level

Figure 2. Large Solution Level

The Solution Train (Figure 3) is the main organizational construct that coordinates the efforts of multiple ARTs and suppliers. The value delivered by Solution Trains can range from core banking applications in global financial institutions to jet fighters and satellite systems. Enterprises that build these systems-of-systems require abilities, principles, and practices beyond those followed by a single ART. Read the [Enterprise Solution Delivery](#) article for further discussion and best practices

for building large solutions with Solution Trains.

Figure 3. SAFe Solution Train

Figure 3. SAFe Solution Train

Large Solution Highlights

Below are the highlights of the Large Solution level:

Solution Train ▷ the key organizational element of the Large Solution SAFe. Aligns the people and the work toward a shared solution vision, intent, and backlog.

Solution Intent ▷ the repository for storing, managing, and communicating the solution's current and intended requirements and design decisions. It provides support for verification, validation, and **Compliance** activities and is an enabler for Lean systems engineering practices, including **Set-Based Design**, **Model-Based Systems Engineering** (MBSE), and **Agile Architecture**.

Solution Train Flow ▷ a state where each Solution Train delivers a continuous flow of value to the customer.

Spanning Palette ḫ the full spanning palette and all its elements, including [Vision](#), [OKRs](#), [Roadmap](#), [AI](#), [Shared Services](#), [Communities of Practice \(CoP\)](#), [System Team](#), and [Measure and Grow](#).

Roles

The Large Solution SAFe roles help coordinate multiple ARTs and suppliers and provide the necessary cross-ART coordination and governance:

[Solution Management](#) ḫ responsible for defining and supporting the building of desirable, feasible, viable, and sustainable large-scale business solutions that meet customer needs over the solution’s significant lifespan.

[Solution Architects](#) ḫ responsible for defining and communicating a shared technical and architectural vision across a Solution Train to help ensure the solution under development is fit for its intended purpose.

[Solution Train Engineer \(STE\)](#) ḫ a servant leader and coach for the Solution Train, facilitating and guiding the work of all ARTs and Suppliers in the Value Stream.

[Supplier](#) ḫ an internal or external organization that develops and delivers components, subsystems, or services, which help Solution Trains deliver solutions to customers.

[Shared Services](#) ḫ represent the specialty roles, people, and services required for the success of an Agile Release Train (ART) or Solution Train, but that cannot be dedicated full-time.

[Communities of Practice \(CoP\)](#) ḫ organized groups of people who have a common interest in a specific technical or business domain who regularly share information, improve their skills, and actively work on advancing the general knowledge of the domain.

Events

Large Solution SAFe highlights events and activities that help coordinate multiple ARTs and suppliers:

[Pre-Plan](#) ḫ describes the alignment activities that prepare Agile Release Trains within a Solution Train for PI Planning.

[Coordinate and Deliver](#) ḫ describes the SAFe artifacts and practices necessary for Solution Trains to maintain alignment and deliver value throughout a PI.

[Solution Demo](#) ḫ integrates the development efforts from all ARTs and suppliers on the Solution Train every PI and makes them visible to Customers and other stakeholders for evaluation and feedback.

[Inspect & Adapt](#) (I&A) ḫ a significant event where the current state of the integrated solution across all ARTs is demonstrated and evaluated. Solution Train stakeholders then reflect and identify improvement backlog items via a structured problem-solving

workshop.

Product Manager Sync Δ similar to the PO Sync (see [PI article](#)), the Product Manager sync typically runs on the same cadence as the PO Sync and is held soon after it to raise and address issues from the ARTs.

Architect Sync Δ a Solution Train event used to guide emerging designs, discuss tradeoffs, and increase opportunities to align implementation approaches without becoming a source of delay.

RTE Sync Δ similar to the ÔCoaches SyncÕ (see [PI article](#)), this event usually runs on the same cadence as the Coaches Syncs and is held soon after it to address issues that canÕt be resolved directly by the ARTs.

Artifacts

The following Large Solution SAFe artifacts help coordinate multiple ARTs and suppliers:

Capabilities Δ a higher-level solution behavior that typically spans multiple ARTs. They are sized and split into various features so they can be implemented in a single PI.

Enabler Capabilities Δ support the activities needed to extend the [Architectural Runway](#) to provide future business functionality, including exploration, architecture, infrastructure, and compliance.

Solution Train Epics Δ epics implemented by a single Solution Train.

Nonfunctional Requirements (NFRs) Δ define system attributes such as security, reliability, performance, maintainability, scalability, and usability and serve as constraints or restrictions on the systemÕs design.

Solution Train Backlog Δ the container for upcoming capabilities and enablers, each of which can span multiple ARTs and is intended to advance the solution and build its architectural runway.

Solution Δ a product, service, or system ARTs deliver to the enterpriseÕs internal or external customers.

Apply SAFe Elements to Other Configurations

SAFe introduces several unique elements in the different configurations. Generally, any SAFe element may be applied to any SAFe configuration. For example, a single ART building a medical device of modest scale will likely have one or more suppliers and a solution intent to manage compliance. Or a Solution Train building a LIDAR system for autonomous vehicles could leverage DevOps. This is part of SAFeÕs scalability and versatility (Figure 4).

Figure 4. Applying SAFe elements to other configurations

Last update: 5 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)
[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

TRAINING

[YouTube](#)

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The most important problem that we face as software professionals is this: If somebody thinks of a good idea, how do we deliver it to users as quickly as possible? [1]

Continuous Delivery

Adjust Cookie Settings

CALMR

CALMR is the **second article** in the **SAFe DevOps series**. It describes the shared mindset and values that support successful DevOps adoption. The following links provide access to the next two articles: the [DevOps home page](#) and SAFe's [DevOps Practice Domains](#).

CALMR is a DevOps mindset that guides the ART toward achieving continuous value delivery by enhancing culture, automation, lean flow, measurement, and recovery.

Successful DevOps hinges on an approach that unites everyone in the value stream toward achieving extraordinary business outcomes. In SAFe, CALMR provides such an approach. When everyone in the value stream thinks and acts with continuous delivery in mind, the result is:

- Increased frequency, quality, and security of product innovation
- Decreased deployment risk with accelerated learning cycles
- Accelerated time-to-market
- Improved solution quality and shortened lead time for fixes
- Reduced severity and frequency of defects and failures
- Improved Mean Time to Recover (MTTR) from production incidents

A critical component of the CALMR mindset is the realization that DevOps often forces significant change within established enterprises. Enterprises are complex systems with diverse people, values, processes, policies, and technology. Careful attention must be given to effectively cultivating and maturing DevOps in these environments.

After more than a decade of experimentation and learning, the DevOps community has discovered that effective DevOps entails a deep appreciation for culture, automation, lean flow, measurement, and sharing (CALMS). DevOps requires directing energy toward each area—not necessarily equally, but in balance—to achieve desired outcomes.

SAFe echoes this belief with one modification, sharing is a natural component of culture, which makes room for “recovery” as a new element. Hence, SAFe’s “CALMR” approach to DevOps (Figure 1).

Figure 1. SAFe's CALMR approach to DevOps

Figure 1. SAFe's CALMR approach to DevOps

CALMR includes five elements that define DevOps excellence. These elements guide the decisions and actions of everyone involved in enabling continuous value delivery.

Details

Culture

In SAFe, DevOps leverages the culture created by adopting the entire Framework's Lean-Agile values, principles, and practices. Every tenet of SAFe, from [Principle #1 - Take an economic view](#) to [Principle #10 - Organize around value](#), applies to DevOps. It enables shifting some operational responsibilities upstream while following development work downstream into deployment and operating and monitoring the solution in production. Such a culture requires:

Customer-centricity □ Value is determined by an enterprise's ability to sense and respond to customer needs; therefore, everyone in the value stream must be guided

by a shared understanding of their customers.

Collaboration DevOps relies on the ability of development, operations, security, and other teams to partner effectively on an ongoing basis, ensuring that solutions are developed, delivered, and maintained in lockstep with ever-changing business needs.

Risk tolerance DevOps requires widespread acknowledgment that every release is an experiment until validated by Customers and that many experiments fail. DevOps cultures reward risk-taking, continuous learning, and relentless improvement.

Knowledge sharing Sharing ideas, discoveries, practices, tools, and learning across teams, ARTs, and the broader organization unifies the enterprise and enables skills to shift left.

Automation

DevOps recognizes that manual processes are the enemy of fast value delivery, high productivity, and safety. Manual processes increase the probability of errors in the delivery pipeline, particularly at scale. These errors, in turn, cause rework, which delays desired outcomes.

Automating the [Continuous Delivery Pipeline](#) (CDP) via an integrated tool chain (Figure 2) accelerates processing time and shrinks feedback cycles. This feedback from customers, stakeholders, solutions, infrastructure, and the pipeline provides objective evidence that solutions are (or are not) delivering the expected value.

Figure 2. A Conceptual CDP toolchain

Figure 2. A Conceptual CDP toolchain

Building and operating a CDP tool chain typically involves the following categories of tools:

Value Stream Management (VSM) □ VSM tools wrap the CDP from end to end, providing real-time visibility into the health and efficiency of the value stream itself.

Version Control □ These tools store and manage changes to source code and configuration files that define the behavior of solutions, systems, and infrastructure.

Infrastructure as code (IaC) □ As a discipline, *infrastructure-as-code* treats all systems as highly configurable, expendable commodities. Tools in this category enable all computing infrastructure to be built, deployed, changed, and destroyed on demand.

Test Automation □ Test automation can be a significant source of delivery acceleration. It applies to almost all testing types, including unit, component, integration, regression, performance, acceptance, usability, and exploratory testing. However, automating exploratory testing requires manual input from those testing the solution.

Vulnerability Detection □ These tools span much of the CDP and are specifically

designed to detect security vulnerabilities in code, networks, and infrastructure.

CI/CD \triangleright [Continuous Integration](#) (CI) and [Continuous Deployment](#) (CD) tools are typically invoked automatically upon code commit and orchestrate build, integration, testing, compliance, and deployment activities.

Monitoring and Analytics \triangleright These tools collect usage and performance data from all levels of the solution stack and provide critical insights into pipeline flow, solution quality, and delivered value.

Additional tools \triangleright The tools above tend to be used universally; however, many others support DevOps but are implementation specific. These include IDE plugins, microservices, artifact repositories, cloud management, and chaos engineering.

Lean Flow

Agile Teams and ARTs strive to achieve a state of continuous flow, enabling new features to move quickly from concept to cash. The key to accelerating flow is reflected in [Principle #6 \$\triangleright\$ Make value flow without interruption](#). Faster flow can best be achieved by adopting all eight *Öflow acceleratorsÖ* described here.

These powerful accelerators of value apply to all Framework levels, but the challenges differ for each. An individual SAFe article discusses how to use these accelerators for each flow domain: [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#). Three of these accelerators, however, are particularly relevant to the ongoing optimization of the CDP. Each is described below in the context of DevOps.

Figure 3. The ART Kanban helps visualize and limit WIP

Figure 3. The ART Kanban helps visualize and limit WIP

Visualize and limit Work in Process (WIP) ☰ Figure 3 illustrates an example of an ART Kanban board, which makes WIP visible to all stakeholders. Kanban boards help teams quickly identify bottlenecks and balance the amount of WIP against the available development and operations capacity.

Work in smaller batches ☰ Small batches go through the system faster and with less variability than large batches. This accelerator supports more frequent deployments and speeds up the learning process. Reducing batch sizes typically involves focusing more attention on and increasing investment in infrastructure and automation that reduces the transaction cost of each batch.

Reduce queue lengths ☰ The size of the queue of work to be done is a predictor of the amount of time it will take to complete the job, no matter how efficiently it is processed. Fast flow is achieved by closely managing and generally reducing queue lengths—the shorter the queue, the quicker the delivery.

Measurement

Achieving extraordinary business outcomes with DevOps requires the CDP to be highly optimized. Solutions, the systems on which they run, and the processes by which they are delivered and supported have to be frequently fine-tuned for maximum performance and value.

The decisions of what to optimize, how to optimize, and how frequently can be guided by [Principle #1](#) **Take an economic view** and [Principle #5](#) **Base milestones on an objective evaluation of working systems**, not merely intuition. The ability to accurately measure delivery effectiveness and feed that information into relentless improvement efforts is critical to the success of DevOps.

The next question is, what metrics should be tracked, and from what sources? While every enterprise and delivery pipeline is somewhat different, the following guidelines apply universally.

Measure pipeline flow \triangleright The health of the delivery pipeline itself can make or break a solution. The [Development Value Stream](#) needs to evolve into a CDP to achieve business agility.

Flow measurements, described in [Measure & Grow](#), focus on throughput and lead time from concept (Customer request) to cash (delivery to the Customer) and are derived from the people and tools that perform design, development, testing, deployment, and release activities. For example:

The **Flow Framework** defines four flow metrics—*flow velocity*, *flow efficiency*, *flow time*, and *flow load*—and tracks the “Flow Distribution” of *features*, *defects*, *risks*, and *debts* in the pipeline.[2]

Google considers **end-to-end lead time** and **deployment frequency** metrics the most crucial pipeline performance indicators. [3]

Measure solution quality \triangleright DevOps cultures stress the importance of shifting technical practices left (earlier). This practice ensures that quality is built into solutions during development rather than “inspected in” as defects are discovered later.

Quality metrics gauge adherence to functional, nonfunctional, security, and compliance requirements, which are best obtained via automated testing tools before release. The Flow Framework categorizes these as *quality* metrics [2], while Google focuses specifically on *change failure* rates. [3]

Measure solution value \triangleright A streamlined pipeline is worthless if it simply accelerates the delivery of products nobody wants. Therefore, measuring the business value of the work exiting the pipeline is essential. These metrics gauge economic outcomes and Customer (or end-user) satisfaction, which are evaluated against forecasted results defined as part of the original business hypothesis.

Value metrics are sourced from full-stack telemetry, analytics engines, financial systems, and feedback from users and stakeholders. The Flow Framework presents these as “business results” with specific metrics to track the *value*, *cost*, and *happiness*.^[2] Google adds *time to restore*—equivalent to the well-known Mean Time to Restore (MTTR)—since production failures can rapidly diminish the value of deployed solutions. [3]

Recovery

It’s imperative to design the CDP for low-risk releases and fast recovery from operational failure to support frequent and sustained value delivery. The Release on Demand article describes techniques for a more flexible release process. In addition, the following practices support fast recovery:

Stop-the-line mentality ▷ With a “stop-the-line” mentality, any issue compromising solution value causes team members to stop what they are doing and swarm on the problem resolution. Learnings are then turned into permanent fixes to prevent the issue from recurring.

Plan for and rehearse failures ▷ When it comes to DevOps, failed deployments can still occur. To minimize and maximize the resiliency of solutions, teams should develop recovery plans and practice them often in production or production-like environments. (See “Chaos Monkey” [4].)

Fast fix forward and roll back ▷ Since production failures are inevitable, teams need to develop the capability to quickly “fix forward” and, where necessary, move back to a known stable state. Fixes must flow through the same process as any feature or enhancement; therefore, the CDP should accommodate any type of change at any level of severity.

Architecture, infrastructure, and skills challenges typically need significant improvement to enable fast, elegant recovery. Organizations often undertake special enterprise-level initiatives to evolve these capabilities.

More in the DevOps Series

Article 1: [DevOps Main Page](#)

Article 2: A CALMR Approach to DevOps (this page)

Article 3: [SAFe® DevOps Practice Domains](#)

Learn More

[1] Humble, Jez, and David Farley. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison-Wesley, 2010.

[2] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. IT Revolution Press, 2018.

[3] Accelerate → State of DevOps 2019. <https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>

[4] The Netflix Simian Army. <https://netflixtechblog.com/the-netflix-simian-army-16e57fbab116>

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

TRAINING

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Inspection does not improve the quality, nor guarantee quality. Inspection is too late. The quality, good or bad, is already in the product. Quality cannot be inspected into a product or service; it must be built into it.

W. Edwards
Deming

Adjust Cookie Settings

Built-In Quality

Built-In Quality is a set of practices to help ensure that the outputs of Agile teams in business and technology domains meet appropriate quality standards throughout the process of creating customer value.

Go

Find a Course:

Details

To support [Business Agility](#), enterprises must continually respond to market changes. The quality of the work products that drive business value directly determines how quickly the teams can deliver their solutions. Although work products vary by domain, they are likely to involve software, hardware designs, scripts, configurations, images, marketing materials, contracts, and other elements. Products built on stable foundations that follow standards are easier to change and adapt. Built-in quality is even more critical for large solutions, as the cumulative effect of even minor defects and wrong assumptions may create unacceptable consequences.

Building quality in requires ongoing training and commitment. But the benefits warrant the investment and include:

- Higher customer satisfaction
- Improved velocity and delivery predictability
- Better system performance
- Improved ability to innovate, scale, and meet compliance requirements

Built-in quality is linked to the fast flow of value described in [SAFe principle 6: Make value flow without interruptions](#). Accelerating problem discovery and taking corrective action occurs by shifting learning left on the timeline. Improved collaboration, workflow automation, more frequent delivery, and faster customer feedback support a quicker learning process.

SAFe applies Built-in Quality across five key domains. Each domain has a set of quality practices that vary from universally applicable generic practices to those specific to one or a few domains.

Figure 1 provides a consolidated view of Built-in Quality in SAFe.

Figure 1. Key domains and practices of Built-In Quality in SAFe

Figure 1. Key domains and practices of Built-In Quality in SAFe

The rest of this article describes the components of Figure 1 in deeper detail.

Built-in Quality Domains

Built-in Quality practices vary based on the domains in which they are applied. Despite the same intent behind the Built-in Quality approach to creating customer value, the actual practices reflect the intricacies of their environment and context. The following are the Built-in Quality domains in SAFe:

Business Functions

Business functions include marketing, sales, HR, finance, supply chain management, and other non-IT disciplines. Along with routine operations, each function also includes complex efforts requiring specific quality outputs for success. For example, creating a new marketing campaign or establishing new HR policies involve certain quality expectations.

Software Applications

Software is an essential contributor to business agility, the ability to scale the business, and better compete in the digital age. But seizing such opportunities requires maintaining predictable quality when delivering solutions.

IT Systems

IT infrastructure powers vast ecosystems of today's enterprise solutions landscape. The more complex the solutions, the more sophisticated the IT systems must be to sustain them. To support the reliable operation of the enterprise, IT systems require substantial quality standards and, therefore, proper quality practices.

Hardware

When used in computer technology, hardware typically refers to cables, monitors, integrated circuits, and other tangible elements of a computer system. But more generally, hardware refers to devices with concrete physical properties: mass, size, and matter. Examples include motors, gears, tools, chassis, cases, and simple or complex mechanisms. Due to their significantly higher cost of change, hardware systems require a unique approach to quality.

Cyber-physical Systems

Cyber-physical systems are complex systems wherein multiple physical elements are controlled by software algorithms. Examples include robots, aircraft, and automobiles. These are some of the world's most complex systems and often include intricate electrical, mechanical, optical, fluidic, sensory, and other subsystems. Their complexity and the high impact of failure emphasize the critical importance of quality in such systems.

Basic Agile Quality Practices

Basic Agile quality practices can be applied to work products in any domain. They have proven their worth and provide a common starting point for knowledge workers to understand and improve the quality attributes of the artifacts, work products, systems, and services that benefit themselves and their customers. A set of five SAFe Basic Agile Quality Practices are described in the sections below.

Shift Learning Left

Every development effort involves numerous unknowns that surface as development progresses and teams learn new facts. If the learning happens late in the process, underlying issues will significantly impact the solution, and significant rework and delays will result. However, if learning takes place much earlier—*or is shifted left*—problems reveal themselves sooner, enabling corrective action with minimum impact (Figure 2).

Figure 2. Late problem discovery vs. shift left

Figure 2. Shifting quality left reveals problems sooner

Shifting learning left does not simply mean that some actions take place earlier on the timeline but also that the structure of some of the basic processes is changed. For example, a test-first approach requires shifting away from conventional testing. Instead, tests are created whenever possible **before** the desired solution functions are implemented.

Pairing and Peer Review

Pair work describes a practice wherein two knowledge workers collaborate over the same asset in real time. Often, one serves as the driver, directly advancing the work product, while the other acts as the navigator, providing real-time evaluation and feedback. Team members switch roles frequently. Because the work product will contain each member's shared knowledge, perspectives, and best practices, pairing creates and maintains higher quality. As teammates learn from each other, the skillsets of the entire team rise and broaden. Additionally, peer review helps spot quality issues as one team member examines the work products of the other. Many governance processes around software, for example, mandate peer review as a compliance activity.

Collective Ownership and T-shaped skills

Collective Ownership is a quality practice where individual team members have the requisite skills and authority to update any relevant asset. This approach reduces dependencies between teams and ensures that any individual team member or team will not block the fast flow of value delivery. Any individual can add functionality, fix errors, improve designs, or refactor *because the work product is not owned by one team or individual*. Collective ownership is supported by quality standards that encourage consistency, enabling everyone to understand and maintain the quality of each component. Collective ownership is further enabled by ÔT-shaped skills. T-shaped skills characterize individuals who possess deep experience in one area but also have broad skills in other areas. T-shaped skills also represent the ability to work well with others.

Artifact Standards and definition of done

Assets created and maintained by the organization must adhere to standards that help ensure their value to the business. These standards may reflect how the artifacts are being built or what properties they must manifest. Standards are often unique to the specific organization and solution context, emerging gradually, validated frequently, and corrected by multiple feedback cycles. To productively maintain artifact standards, the teams must understand the motivations for their existence. Artifact design practices and the effective use of automation help facilitate standards. Enacting productive artifact standards involves applying a definition of done (DoD) Ô an essential way of ensuring that a work product is complete and correct. Each team, train, and enterprise should build a DoD that suits their needs.

Workflow Automation

Workflows tend to have many manual steps. Handoffs from one worker to another, searching for an asset of interest, and manual inspection of an asset to a standard are just a few examples. The fact is, all these manual steps are error-prone and cause delays in the process. Many of these tasks can be automated if the teams take the time to invest in a more automated pipeline that supports the activities. Automation provides substantial gains due to reduced execution costs and intrinsic adherence to standards. Of course, this can be done incrementally, and it often starts by putting a Kanban system in place and then noting steps that can be automated. Sometimes the first step is simply setting up automated notifications when an item changes state. Even simpler, many such systems are designed as true pull systems where the worker simply checks the system to see what work is available to them based on its state. In this case, the handoff is automatic and doesnÔt require separate communications overhead just to know the state of a work product.

Business Quality Standards

The above sections describe a set of five basic Agile quality practices that can be applied to every business domain. Virtually every aspect of business operations—accounting and finance, legal, sales, development, HR, marketing, operations, production, and more—is subject to internally or externally imposed quality standards, which are often linked to compliance requirements. Each business function produces specific outputs, which must satisfy quality standards relevant to that context.

No matter your business function, the steps to achieve quality with Agility include the following:

- Organize into Agile teams, get trained, and iterate.
- Define the standards and compliance policies for your function.
- Agree on the definition of done (DoD) for artifacts and activities for your workflow.
- Implement the basic Agile quality practices.
- Measure and learn. Specialize Agile quality practices further to your specific function.
- Improve relentlessly.

Agile Software Development Quality Practices

Software may well be the richest and best-defined area for applying Built-in Quality. This was driven by necessity, as software is exceedingly complex and intangible. You can't touch it or see it, so traditional approaches to inspecting, measuring, and testing are inadequate. If quality isn't built in endemically, then it's unlikely to exist at all. To address this new challenge, many new quality practices like those above were

inspired by Extreme Programming (XP), which has a zest for going fast

with quality. They have proven their worth and have now started influencing quality practices in other domains. The practices below apply well to software development, and we will describe them in that context, but they can be applied to other domains as well.

Continuous Integration

Building large-scale value requires knowledge workers to build the system in increments, resulting in frequent small changes. Each must be continually checked for conflicts and errors and integrated with the rest of the system to assure compatibility and forward progress. [Continuous Integration](#) (CI) provides developers with fast feedback (Figure 3). Each change is quickly built, integrated, and then tested at multiple levels. CI automates the process of testing and migrating changes through different environments, notifying developers when tests fail.

Figure 3. Continuous integration (CI) fosters system-wide quality

Figure 3. Continuous integration (CI) fosters system-wide quality

Continuous integration is vital within and across teams, allowing them to quickly identify and resolve issues in all parts of the codebase.

Test-first Practices

Agile teams operate in a fast, flow-based system to develop and release high-quality business capabilities quickly. Instead of performing most of the testing at the end, Agile teams define and execute many tests early and often as a part of their integration process. Tests are defined for small units of code using [Test-Driven Development](#) (TDD), for [Story](#), [Feature](#), and [Capability](#) acceptance criteria using [Behavior-Driven Development](#) (BDD), and for the feature or capability benefit hypothesis using [Lean UX](#) (Figure 4). Building quality in ensures that Agile development's frequent changes do not introduce new errors while enabling fast, reliable execution.

Figure 4. Test-first practices accelerate flow

Figure 4. Test-first practices accelerate flow

Refactoring

Constantly changing technology and evolving business objectives make it difficult to maintain and continually increase business value. However, two paths to the future exist:

Keep adding new functionality to an existing code base toward an eventually unmaintainable “throw-away” state

Continuously refactor the system to build a foundation for efficiently delivering the current business value as well as future business value

Refactoring, which improves the internal structure or operation of an area of code without changing its external behavior, is better. With continuous refactoring, the useful life of an enterprise’s investment in software assets can be extended substantially, allowing users to benefit from a flow of value for years to come. But refactoring takes time, and the return on investment is not immediate, so an allowance for time and effort must be part of capacity planning considerations. For

more, see the extended guidance article on [Refactoring](#).

Continuous Delivery

Continuous delivery provides the ability to release value to customers whenever they need it. This is accomplished by the [Continuous Delivery Pipeline](#) (CDP), which contains four aspects: *continuous exploration*, *continuous integration*, *continuous deployment*, and *release on demand*. The CDP enables organizations to map their current pipeline into a new structure and use relentless improvement to deliver value to customers. Feedback loops internally within and between the steps and externally between the customers and the enterprise fuel improvements. Internal feedback loops often center on *process* improvements; external loops often center on *solution* improvements. The improvements collectively create synergy, ensuring the enterprise is Ôbuilding the right thing, the right wayÕ and frequently delivering value to the market. Additionally, SAFe [DevOps](#) features crucial practice domains for establishing fast and reliable value delivery mechanisms.

Continuous delivery helps SAFe teams release on demand. Releasing with quality, however, requires a specific, scalable definition of done that helps ensure that the requisite quality is built in. Figure 5 shows an example:

Figure 5. An example of a scalable definition of done

Figure 5. An example of a scalable definition of done

To support security practices, teams generate a Software Bill of Materials (SBOM) for each release describing the commercial and open-source components and dependencies to ensure no vulnerabilities.

Agile Architecture

[Agile Architecture](#) is a set of values, practices, and collaborations that support a system's active, evolutionary design and architecture. It embraces the DevOps mindset, allowing a system's architecture to evolve continuously while simultaneously supporting the needs of current users.

Agile architecture supports Agile development practices through collaboration, emergent design, intentional architecture, and design simplicity. It also enables designing for testability, deployability, and changeability. Rapid prototyping, set-based design, domain modeling, and decentralized innovation, in turn, support Agile architecture.

The essential concept of [Architectural Runway](#) allows Agile teams and trains to provide effective enablement for future business capabilities and features while progressively validating underlying architectural assumptions.

IT Systems Quality Practices

Every modern enterprise depends on properly functioning IT systems for its business success. With more and more business workflows being powered by IT, ensuring the reliability, scalability, safety, and security of IT systems becomes increasingly important. It requires a robust approach to building quality into these systems. A sample of IT-specific quality practices is described below.

Infrastructure as Code

One of the critical challenges in ensuring the quality of IT ecosystems comes from defining and sustaining configurations consistently. Often representing hundreds or even thousands of environment parameters, configurations grow out of sync and cause problems in different parts of the enterprise's solution landscape. "Infrastructure as Code" is an approach to control those configurations programmatically and thus benefit fully from automation in defining, procuring, and maintaining configurations consistently and integrally. Containerization is an excellent enabler of Infrastructure as Code, as it permits applying programming interfaces to various aspects of the execution environment. Additionally, using "immutable infrastructure" – an approach where IT components are rebuilt whenever needed, rather than modified in production – forces the organization to explicitly control all changes to the environment by formally redefining them and redeploying the component that changed.

NFRs and SLAs

IT infrastructure must provide certain qualities to the execution environment to support the systems essential to business operation. These quality attributes include things such as security, reliability, performance, maintainability, and scalability (Nonfunctional Requirements or NFRs). Additionally, relevant Service-Level Agreements (SLAs), such as Mean Time Before Failure (MTBF) and Mean Time to Repair (MTTR), must be ensured. In SAFe, NFRs and SLAs are achieved

incrementally by early and continuous testing and timely corrective action. Ensuring that systems meet their NFRS and SLAs requires instrumentation and the proactive build and use of the architectural runway.

Telemetry and Monitoring

Responding to unanticipated loads, security attacks, hardware, software, and network failures, require a range of options, from downgrading or removing services to adding service capacity. Telemetry and logging capabilities allow organizations to understand and fine-tune their architecture and operating systems to meet intended loads and usage patterns. Effective monitoring requires that full-stack telemetry is active for all features deployed through the CDP. Monitoring ensures that issues with system performance can be anticipated or addressed rapidly in production.

Cybersecurity Standards

IT environments must meet increasingly stringent quality standards to protect against unauthorized access, use, disclosure, or destruction. The spectrum of activities to achieve comprehensive cybersecurity includes:

- Technology enablement (data encryption, streamlined identity management, etc.)
- Frequent testing and validation (audits, penetration testing, etc.)
- Training and proper habits for the workforce
- Testing of all new assets for various vulnerabilities
- Frequently review new vulnerability alerts against existing solution's SBOM for affected components and provide patches or hotfixes

Automated Governance

Recent advances in DevOps and related methods, practices, and tooling provide new opportunities for IT teams to automate governance. Automated governance replaces tedious, manual, and error-prone activities and specifically addresses security, compliance, and audit needs. For more on this topic, see the reference [2]: Investments Unlimited, A Novel about DevOps, Audit Compliance, and Thriving in the Digital Age.

Automation of configuration management, audit, security testing (during both build and deployment), and immutable infrastructure help reduce human error that can lead to system vulnerabilities.

Agile Hardware Engineering Quality Practices

Ensuring quality in hardware systems and components is complicated because the cost of change increases with time, and the impact of quality issues with hardware is high. This can include catastrophic field failure, recalls of volumes of manufactured products, and expensive field replacement or repair. This risk pressures organizations to effectively apply Built-in Quality practices while developing engineered hardware systems and subsystems. There are several techniques organizations use to ensure Built-in-Quality in hardware systems, which are described below.

Modeling and Simulation

In Agile, the goal is to build and learn as quickly as possible. Modeling and simulation in the virtual environment and rapid modeling in the prototype environment help shift learning left, as shown in Figure 6.

Figure 6. Shift learning left for hardware engineering

Figure 6. Shift learning left for hardware engineering

Analysis and simulation in digital models used in electrical and mechanical Computer-Aided Design (CAD) and MBSE (see below) can test changes quickly and economically. Digital twins combine multiple virtual models with data harvested from telemetry in the operational systems to improve the models and better predict how systems will behave in the future. The feedback loops in Figure 6 show how data from other environments validate and improve the digital environment. Some aerospace and automotive products even use model simulations for certification, substantially reducing the time and cost of changes.

Rapid Prototyping

The virtual environment cannot reveal all issues. Physical prototypes are a lower-cost substitute for real, Òbent metalÓ hardware. They provide higher-fidelity feedback, available only in a physical environment. Example prototype practices include:

Wood and other low-fidelity mockups

Breadboarding electrical components
3d-printed mechanical and electrical parts (PCBs, wiring harnesses)

Increasingly, additive manufacturing is used to lower the costs of rapid experimentation and prototyping. Additive manufacturing uses data computer-aided-design (CAD) software or 3D object scanners to direct hardware to deposit material, layer upon layer, in precise geometric shapes. As its name implies, additive manufacturing adds material to create an object. By contrast, when you create an object by traditional means, it is often necessary to remove material through milling, machining, carving, shaping, or other means.^[3]

Many organizations with the equipment and knowledge to print mechanical and electrical parts can produce and ship them in a single day. And parts made with additive manufacturing are now making their way into production.

Cyber-physical Systems Quality Practices

Cyber-physical systems require an organization to deal effectively with hardware components and the software that governs its behavior. Additionally, because such systems operate directly in the real world, the impact of quality issues can be significant and often subject to regulatory compliance.

Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE) is the practice of developing a set of related digital models that help define, design, and document a system under development. These models provide an efficient way to explore, update, and communicate system aspects to stakeholders while significantly reducing or eliminating dependence on traditional documents. By testing and validating system characteristics early with the model, they facilitate timely learning of properties and behaviors, enabling fast feedback on requirements and design decisions.

Frequent End-to-end Integration

In the software domain, continuous integration is the heartbeat of continuous

delivery: It's the forcing function that verifies changes and validates assumptions across the entire system. Agile teams invest in automation and infrastructure that builds, integrates, and tests every developer change, providing immediate feedback on errors.

Large, cyber-physical systems are far more challenging to integrate continuously because:

- Long lead-time items may not be available
- Integration spans organizational boundaries
- Automation is rarely end-to-end
- The laws of physics dictate certain limitations

Instead, *frequent* end-to-end integration addresses the economic tradeoffs of the transaction cost of integrating versus delayed knowledge and feedback (Figure 7).

Figure 7. Frequent end-to-end integration

Figure 7. Frequent end-to-end integration

The goal is frequent partial integration with at least one complete solution

integration for each PI.

Learn More

[1] https://en.wikipedia.org/wiki/Software_supply_chain

[2] Beal, Helen and Bill Bensing , Jason Cox , Michael Edenzon , John Willis.

Investments Unlimited: A Novel about Devops, Security, Audit Compliance, and Thriving in the Digital Age, IT Revolution Press, 2022

[3] <https://www.ge.com/additive/additive-manufacturing>

Last updated: 18 January 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Applying
management
frameworks from a
hundred years ago
to organizations that
need to compete in
the digital age is
futile.*

From *Project to
Product* Ñ Mik
Kersten [1]

Adjust Cookie Settings

Principle #10 Ø Organize around value

As described in the [Business Agility](#) article, successful enterprises don't start as large and cumbersome. Typically, they begin as a network of people cooperating to address customer needs. Structure and formality are unnecessary. People just naturally do whatever needs to be done. Focusing on the customer is natural, too, because without that, Darwinism takes hold and business failure is quick and catastrophic.

But as the enterprise grows and experiences success, it builds the organizational Hierarchy needed to provide the time-tested management structures that will support operations, HR, finance, governance, and all the other corporate responsibilities of a functioning enterprise.

Over time, however, the work of maintaining the Hierarchy overtakes the work of the entrepreneurial Network. The result can be a seemingly decreasing ability to deliver the right products to the right customer at the right time. And success is no longer assured.

While leveraging the hierarchical system benefits and stability, the dual operating system model of Business Agility (Figure 1) restores the entrepreneurial Network's speed and innovation.

Figure 1. The dual operating system of Business Agility

The Network is optimized for speed and adaptability; the Hierarchy is optimized for efficiency and stability. Both are necessary and valuable, and both must be Agile. But they are designed and optimized for different purposes.

SAFe implements the Network as a set of *development value streams* (DVSs) and provides the necessary interfaces to the Hierarchy to restore the system's balance.

Freed from the reporting structure and the need for start-and-stop projects, the Network can now organize and reorganize *as needed*. Moving the focus from Ôproject to productÕ [1] redisCOVERS the customer focus and adaptability required to survive in the digital age. The Network optimizes the flow of value by:

- Reducing handoffs and delays between functional areas, reducing time to market
- Bringing together all the research, development, deployment, and service personnel needed to offer whole product solutions
- Providing intense customer focus across all disciplines for each product and service type
- Measuring success via meaningful, outcome-based key performance indicators
- And perhaps most importantly, the Network can rapidly *reorganize* as necessary to support emerging opportunities and competitive threats

How SAFe Organizes Around Value

This principle, *Principle 10-Organizing around value*, describes how applying this second operating system frees the enterprise to arrange itself to optimize value delivery. It accomplishes this in three nested parts:

1. Build technology portfolios of development value streams
2. Realize value streams with product-focused Agile Release Trains (ARTs)
3. Form Agile teams that can directly deliver value

Each is described in the sections that follow.

Build Technology Portfolios of Development Value Streams

The key to unlocking this potential is understanding and applying value streams, which are fundamental to lean thinking, as described in the [Lean-Agile Mindset](#). Lean thinking can be summarized as follows: [2]

- Precisely specify value by specific product
- Identify the value stream for each product

- Make value flow without interruptions
- Let the customer pull value from the producer
- Pursue perfection

This mindset assures product and customer focus, as the value stream, product, and customer are inexorably linked. You can't have one without the others.

Value streams are defined by the steps, the people, and the flow of information and material necessary to deliver customer value. Value streams optimize the flow of value across divisions and functional departments and through the *system as a whole* to the customer.

SAFe describes two types of value streams, operational and development. While the structure of operational value streams varies significantly based on the purpose and type of value delivered, the structure of development value streams has a standard form, as summarized in Figure 2.

Figure 2. Structure of a development value stream

Figure 2. Structure of a development value stream

Each SAFe portfolio consists of a collection of development value streams aligned as necessary to deliver the products and services customers need (Figure 3). This allows the entire organization—from the building block of Agile teams to ARTs and Solutions Trains, to the entire portfolio—to organize for one purpose: delivering value to the customer as quickly as possible.

Figure 3. A SAFe portfolio is a collection of development value streams

Figure 3. A SAFe portfolio is a collection of development value streams

Organizing a portfolio this way offers many benefits:

- Helps ensure customer and product focus across the entire portfolio
- Aligns strategy to execution by bringing visibility to *all* the work
- Provides the basis for Lean Budgets, which eliminates the friction and cost accounting overhead of traditional project-based work
- Supports measuring success via outcome-based key performance indicators (KPIs)
- Improves workflow with smaller batch sizes

Indeed, when you start to understand the value they bring to the enterprise, it makes you wonder how we got along without value streams. Yes, they were always

there; *we just didn't see them.* [2]

Realize Value Streams with Product Focused Agile Release Trains

The straightforward structure of development value streams begs the question: How do we reduce the time from feature request to delivery? People and resources need to be organized to minimize handoffs, make the individual steps more efficient, and reduce the delays between them. That is the purpose of the ART, as Figure 4 illustrates.

Figure 4. Agile Release Trains realize Development value streams

Figure 4. Agile Release Trains realize Development value streams

ARTs are cross-functional, cross-discipline teams of up to 150 people. To minimize handoffs and delays—and to foster continuous knowledge growth—ARTs have all the business and technical capabilities needed to define, implement,

validate, deploy, release and support solutions for their customers.

Scaling the ART into Solution Trains

In many circumstances, a single ART can deliver and support a significant product or service. When the solution gets bigger, however, the organization gets bigger, too. In this case, ARTs are assembled into Solution Trains capable of building ever-larger systems. To assist with this, SAFe suggests constraining the design options by applying three specific types of ARTs: stream-aligned, complicated subsystem, and platform, as Figure 5 illustrates.

Figure 5. Solution Train example containing two streams aligned ARTs, one platform ART, and one complicated subsystem ART

Form Cross-Functional Agile Teams That Directly Deliver Value

At the heart of this structure are the Agile Teams who build the system, the basic building block of all things Agile. They constitute the majority of personnel on ARTs, and their structure is well defined: each is a cross-functional group of typically ten or fewer individuals who can define, build, test, and deploy an increment of value in a short time box. Like the ART, Agile teams are customer-focused, cross-functional, and have all the skills needed to deliver beginning-to-end value to the customer. (Figure 6)

Figure 6. Agile Teams are cross-functional and organized to deliver value

Figure 6. Agile Teams are cross-functional and organized to deliver value

To limit cognitive load, Agile Teams are organized around one of four types of value: stream-aligned, complicated subsystem, platform, and enabling teams [3] (Figure 7).

Figure 7. Agile Release Train example contains three stream-aligned teams, one platform team, one enabling team, and one complicated subsystem team

Most ART teams are stream-aligned, empowered, and capable of delivering value to their customers with minimal handoffs, delays, and dependencies with other teams. Other teams support stream-aligned teams in their mission.

(Note: For more on team and ART topologies, see [Agile Teams](#), [Solution Train](#), [Organizing Agile Teams and ARTs](#), and reference [4].)

Reorganizing around Value

This principle highlights three specific organizational patterns—Value Streams, ARTs, and Agile Teams—that SAFe enterprises use to build the second operating system. This Network is far more adaptable and can respond more quickly to market changes than the hierarchical system. People and teams can flex naturally to the demands of the incoming work without disturbing any reporting or other relationships in the hierarchical system. And yet, even in a sea of constant change,

the standardization of these patterns adds structure, focus, and stability for the people who do this critical work.

Value streams, ARTs, and Agile Teams live for as long as the solutions they develop and support thrive in the marketplace. But they are not fixed for all time. Some new value streams will be created, others will need to be adjusted, and some will be eliminated as solutions are decommissioned. In other words, when necessary, the Network can evolve quickly while the Hierarchy can remain relatively stable. (Figure 8)

Figure 8. The Network can evolve quickly to address emerging threats and opportunities

Fortunately, the people and teams of an increasingly Lean and Agile enterprise see those changes coming through the portfolio. As virtual organizations, they can quickly evolve in any combination necessary to respond to changing technologies, products, and market demands. In other words, as value moves, these organizations can move with it. Business Agility is the most important outcome.

Learn More

[1] Kersten, Mik. *Project to Product*. IT Revolution Press, 2018.

[2] Rother, Mike, and John Shook. *Learning to See: Value Stream Mapping to Create Value and Eliminate Muda*. Lean Enterprise Institute, 2018.

[3] Skelton, Mathew, and Manuel Pais. Team Topologies: *Organizing Business and Technology Teams for Fast Flow*. IT Revolution Press, 2019.

[4] [Organizing Agile Teams and ARTs: Team Topologies at Scale](#)

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

[About Certification](#)

Weekdays: 9am to 5pm

[Become a Trainer](#)

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Perfection is not attainable, but if we chase perfection, we can achieve excellence.

~Vince Lombardi

Value Stream Management

Value Stream Management (VSM) is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle.

Today, delivering digitally enabled solutions in large enterprises is often complex and spans many functional boundaries. Consequently, this structure can result in a fragmented delivery process with handoffs, cross-team dependencies, breakdowns in communication, and substantial delays. The purpose of VSM is to bring order to this chaos so that value-producing work can flow smoothly and continuously across the organization. The key to succeeding with VSM in SAFe is applying Lean thinking principles to every value stream.

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of eight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Details

In SAFe, Lean thinking is the foundation of VSM. Lean is an extensive body of knowledge aimed at improving operational efficiency by eliminating causes of delay. Lean thinking is based on the following five principles. [1]

1. Precisely specify value by specific product
2. Identify the value stream for each product
3. Make value flow without interruptions
4. Let the customer pull value from the producer
5. Pursue perfection

These principles provide a shared mindset for everyone involved in solution delivery. The remainder of this article describes each of the principles, how they enable VSM in SAFe, and who in the organization is primarily responsible for applying them.

1. Precisely Specify Value by Specific Product

This first principle of Lean thinking underscores the importance of understanding customers' needs and quantifying the value inherent in the products and services delivered. That value comes in the form of enduring [Solutions](#)—not temporary projects or initiatives—and is ultimately determined by the customer.

SAFe [Principle #1](#) **Take an economic view** informs VSM to define an overarching financial strategy for the value stream, including a bi-directional value exchange for each solution delivered (Figure 1).

Figure 1. Solutions require bi-directional value exchange

Figure 1. Solutions require bi-directional value exchange

Digitally enabled solutions are delivered through a SAFe [Portfolio](#) via one or more [Development Value Streams \(DVS\)](#) to external or internal customers. Those solutions support the delivery of market-facing products or services via [Operational Value Streams \(OVS\)](#). Each customer in the end-to-end flow of value derives tangible

benefits from the solutions they receive, such as increased productivity, customer satisfaction, or delight. Likewise, value is returned to the enterprise through cost savings or revenue. Although VSM can be applied to all value streams, SAFe uses VSM primarily to optimize the DVS.

The process of specifying value is aided by [Customer Centricity](#) and [Design Thinking](#). Being acutely tuned to the wants and needs of customers—and having a process to obtain continuous market feedback on solution ideas—ensures products and services deliver mutual, ongoing value.

Determining precisely how much value solutions deliver requires objective measurement. In SAFe, [Key Performance Indicators \(KPIs\)](#) are used to quantify the value solutions provide customers, and the economic benefit returned to the enterprise.

2. Identify the Value Stream for Each Product

Once the value of their products and services has been precisely defined, organizations must identify how they will develop and deliver that value to customers.

A DVS, or simply value stream in the context of this article, is the set of steps an organization executes to deliver a solution to the customer. The sequence of steps is unique to each value stream but generally involves incrementally defining, building, validating, and releasing functionality, as shown in Figure 2.

Figure 2. A basic development value stream in SAFe

Figure 2. A basic development value stream in SAFe

Development value streams are organized into SAFe portfolios, as Figure 3 illustrates. Each value stream delivers one or more solutions—*in the form of products and services*—to the customer. Budgets are allocated directly to value streams, funding the people, systems, and materials required to deliver those solutions fully.

Figure 3. A SAFe portfolio consists of one or more development value streams

Figure 3. A SAFe portfolio consists of one or more development value streams

SAFe Principle #2 [Apply systems thinking](#) guides the organization in identifying an entire value stream from request to release and regarding it as a single, integrated delivery system. This aligns efforts across the organization toward managing one shared value stream instead of many independently operated fragments.

Once the value stream has been identified, all people involved in its steps are organized to ensure smooth execution. Principle #10, [Organize around value](#) requires people with different skills to come together as a unified, enduring, cross-functional [Agile Release Train \(ART\)](#) to build the solution, as illustrated in Figure 4.

Figure 4. An ART contains all the people who work in the value stream

Figure 4. An ART contains all the people who work in the value stream

Large solutions—such as aircraft, automobiles, satellites, and medical imaging devices—typically require multiple ARTs and external suppliers to realize the value stream effectively. [Solution Trains](#) are established to organize the hundreds, sometimes thousands, of individuals needed to manage these complex value streams.

3. Make Value Flow without Interruptions

Once value streams are identified and teams are organized around the solutions they support, the value stream must be cleared of wasteful activities that delay solution delivery. Locating the sources of these delays is the first step toward improving value flow.

Value stream mapping accomplishes this by modeling the end-to-end sequence of activities and measuring performance at and between each step. As shown in

Figure 5, value stream maps help visualize the value stream and pinpoint major delay occurrences.

Figure 5. Value stream mapping identifies sources of delay

Figure 5. Value stream mapping identifies sources of delay

The next step toward accelerating flow is identifying the root causes of these delays and applying Lean-Agile methods to correct them.

SAFe identifies eight flow accelerators that help reduce interruptions and enable continuous value flow. They are listed below and are defined in detail in [Principle #6](#)
D Make value flow without interruptions:

- Visualize and limit WIP
- Address bottlenecks
- Minimize handoffs and dependencies
- Get faster feedback
- Work in smaller batches
- Reduce queue length
- Optimize time Ôin the zoneÕ

Remediate legacy policies and practices

These flow accelerators are time-tested approaches to addressing flow issues. How they are best applied depends on the nature of the issue and where it occurs in the value stream. The [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#) articles provide specific guidance for applying these accelerators at those levels to enable uninterrupted flow across the value stream.

4. Let the Customer Pull Value from the Producer

Value streams must deliver the right value to customers at the right time. This fourth principle of Lean thinking guides organizations to provide solutions that customers pull into the market based on their actual needs rather than pushing solutions based on what they think customers need (Figure 6).

Figure 6. Pull-based vs. push-based value flow

Figure 6. Pull-based vs. push-based value flow

Traditional, push-based delivery systems produce large, infrequent releases that typically deliver too little value too late. The solution is to create a pull-based flow, where small batches of work are prioritized and delivered quickly based on customer feedback and KPIs.

A simple yet powerful way to implement pull-based delivery is with a Kanban system. These are visual representations of the value stream (or segments of the value stream) that help regulate efficient flow through the system, as shown in Figure 7.

Figure 7. Kanban systems foster pull-based processing

Figure 7. Kanban systems foster pull-based processing

Kanban systems are inherently visual, pull-based and create smaller batches by applying WIP limits. SAFe encourages using [Team](#), [ART](#), [Solution Train](#), and [Portfolio](#) Kanban systems.

Further, solutions should be architected to support pull-based flow and on-

demand releases. Design these solutions to be loosely coupled, enabling individual components and sub-systems to be changed, tested, and deployed independently. This strategy avoids the high transaction costs and delays of Ôbig bangÕ change efforts. Accomplish this by architecting for [DevOps](#) and continuous delivery, using domain-driven design (DDD), containerization, and cloud-first methods. Also, feature toggles and blue/green infrastructure patterns effectively decouple deployment from release and allow releasing functionality on demand.

5. Pursue Perfection

VSM is an ongoing practice that continually optimizes the value stream in pursuit of maximum flow and quality. Although a perfectly efficient value stream is not achievable, the constant drive for perfection is a trademark of Lean thinking. It establishes a culture of continuous value stream innovation throughout the organization.

An organization pursuing delivery perfection must regularly assess its value stream against defined performance targets. Proper metrics, both quantitative and qualitative, provide a foundation for effective decision-making, replacing opinions with facts. [Flow metrics](#) (Figure 8) offer a comprehensive view of the work moving through the value stream, along with how quickly, predictably, and efficiently that work is flowing. These objective measurements help to identify the precise locations of delivery bottlenecks.

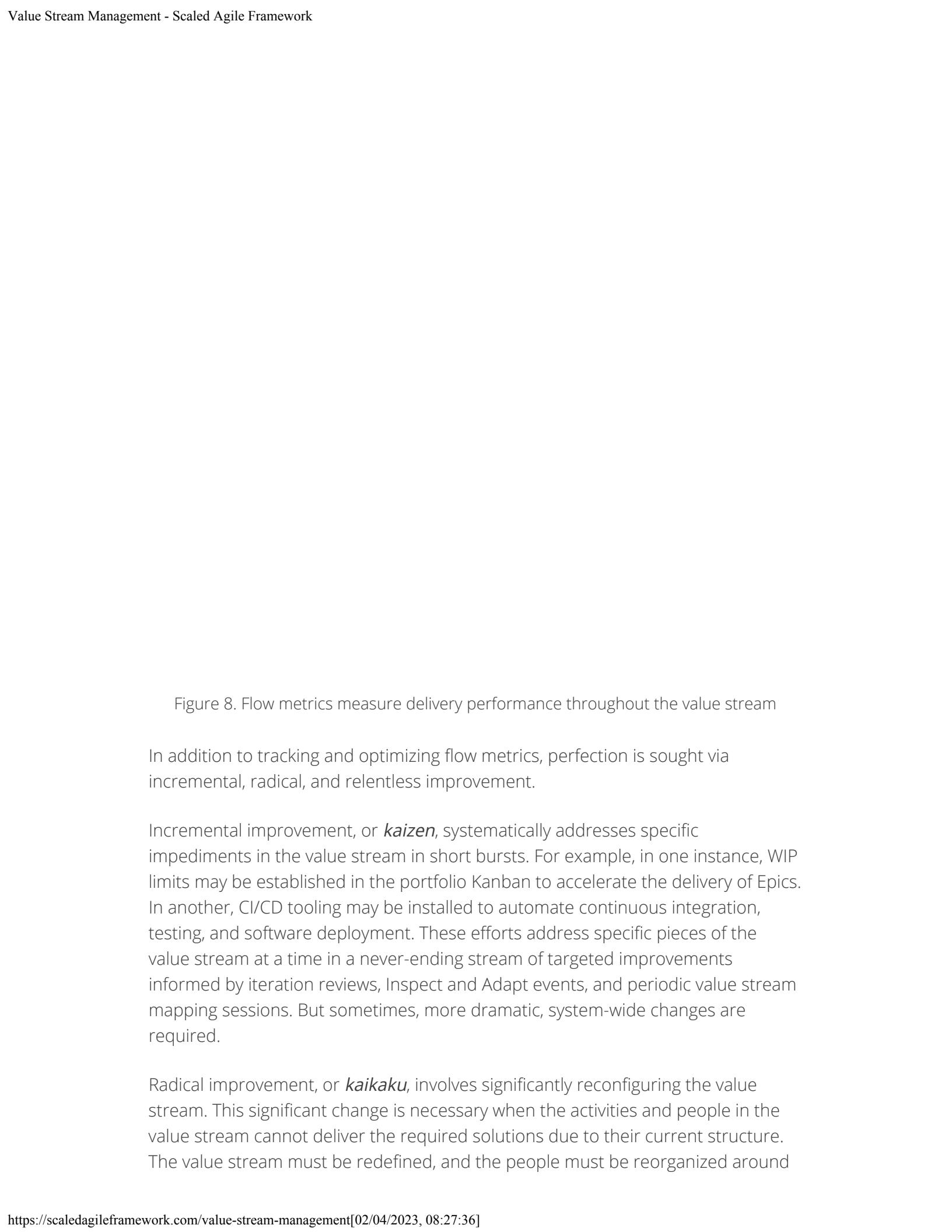


Figure 8. Flow metrics measure delivery performance throughout the value stream

In addition to tracking and optimizing flow metrics, perfection is sought via incremental, radical, and relentless improvement.

Incremental improvement, or *kaizen*, systematically addresses specific impediments in the value stream in short bursts. For example, in one instance, WIP limits may be established in the portfolio Kanban to accelerate the delivery of Epics. In another, CI/CD tooling may be installed to automate continuous integration, testing, and software deployment. These efforts address specific pieces of the value stream at a time in a never-ending stream of targeted improvements informed by iteration reviews, Inspect and Adapt events, and periodic value stream mapping sessions. But sometimes, more dramatic, system-wide changes are required.

Radical improvement, or *kaikaku*, involves significantly reconfiguring the value stream. This significant change is necessary when the activities and people in the value stream cannot deliver the required solutions due to their current structure. The value stream must be redefined, and the people must be reorganized around

the new flow of value. In these circumstances, organizations go through the process of identifying value streams and ARTs, replacing existing activities, and reconfiguring ARTs as necessary.

Organizations must continually identify inefficiencies in the value stream and make incremental and radical improvements when necessary to correct them. This persistent quest for perfection and bias for action signifies relentless improvement — a fundamental component of the Lean-Agile mindset.

Value Stream Leadership in SAFe

VSM is not easy. It entails the disciplined, ongoing application of the five principles of Lean thinking throughout each value stream. This, of course, requires the time, expertise, and dedication of key individuals.

Many organizations have appointed a Value Stream Manager—a role inherited from manufacturing—to serve in this capacity. A Value Stream Manager has the following characteristics:

Lean mindset □ They understand the principles of Lean thinking and how to apply them to improve value stream flow.

Business knowledge □ They understand the customer needs, market forces, and compliance factors that shape product strategy and define value metrics that guide product delivery.

Technical knowledge □ They understand which products, services, and supporting tools will produce the solutions with the most business value.

Process knowledge □ They understand the sequence of activities across the organization that turns ideas into valuable solutions.

Strategic influence □ They evangelize, support, enable, and secure funding for value stream improvements.

Tactical influence □ They identify needed improvements, mobilize teams, lead change, and regularly measure results.

A single Value Stream Manager may suffice for small, independent value streams; however, at scale, it is unlikely that any individual possesses all the necessary knowledge and skills. Therefore, in SAFe, a few key roles typically have joint responsibility for managing the value stream. As illustrated in Figure 9, ART and Solution Train triads carry out the tactical, output-oriented aspects of VSM, while **Business Owners** provide the strategic, outcome-oriented aspects. Together, they act as a team to provide the multi-faceted leadership required to manage the execution and evolution of value streams at scale.

Figure 9. Shared value stream leadership in SAFe

Figure 9. Shared value stream leadership in SAFe

In cases where a value stream manager *has* been designated, the person in this role would likely also serve as a Business Owner for one or more ARTs and assume the strategic VSM responsibilities listed above. If not, at a minimum, they would share the responsibility of managing the value stream with the individuals in these defined SAFe roles.

Triad Responsibilities

Triads align key individuals who possess business, technical, and SAFe expertise to a shared purpose of managing efficient delivery execution. [Product Management](#), [System Architects](#), and the [Release Train Engineer](#) (RTE) jointly guide ART execution, while [Solution Management](#), [Solution Architects](#), and the [Solution Train Engineer](#) (STE) jointly guide Solution Train execution.

Each role carries specific responsibilities within its domain of expertise—*as explained in its Framework article—but together, they have the added collective*

duty of managing the tactical execution of the value stream. Each triad applies the principles of Lean thinking to optimize its value stream for the most desirable outputs, as measured by SAFe's flow metrics.

Members of the triad work together, combining their strengths and spheres of influence, to define and architect solutions, technology stacks, and processes for continuous value flow, facilitate the identification of bottlenecks, and lead the organization through changes that accelerate value delivery.

Business Owner Responsibilities

Business owners are ultimately accountable for value stream outcomes. They provide the vision, strategy, governance, and leadership alignment required to operate and change value streams in alignment with evolving business needs.

Business owners apply the five principles of Lean thinking to optimize their value streams for the most desirable business outcomes. They define the KPIs that steer solution design toward realizing tangible economic benefits. Business owners then guide and support ARTs and Solution Trains through delivery in alignment with these KPIs.

To support relentless improvement, Business Owners sponsor value stream improvement opportunities identified by ART and Solution Train triads. When bottlenecks are so severe that they require radical changes to the value stream, Business Owners provide the strategic influence needed to define, prioritize, and implement systemic—sometimes disruptive—improvements that yield the greatest value to customers.

Summary

VSM is a leadership and technical discipline that enables the maximum flow of business value through the end-to-end solution delivery life cycle. SAFe enables VSM through its alignment with the five principles of Lean thinking and practices that optimize delivery efficiency across the organization.

VSM requires active, ongoing effort by leaders in the organization with business, technical, and process expertise, as well as strategic and tactical influence. In SAFe, these responsibilities are shared among the ART triad, Solution Train triad, Business Owners, and Value Stream Managers (if present) to ensure effective value stream management at scale.

Learn More

[1] Womack, James P., and Daniel T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York, NY: Free Press, 2003.

Last update: 27 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



When we start thinking about ways to line up all of the essential steps needed to get a job done into a steady, continuous flow, it changes everything.

Ñ James P. Womack
and Daniel T. Jones,
Lean Thinking

Adjust Cookie Settings

Principle 6 ☰ Make Value Flow

without Interruptions

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of Ÿight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Enterprises must respond quickly to market changes to remain competitive in the digital age. Delivering a continuous flow of value to customers in the Ÿshortest sustainable lead time is the central theme of SAFe. Doing so requires moving new system features through the development value stream as quickly as possible. Achieving continuous flow requires a new way of working that eliminates the traditional start-stop-start project cycle and the waterfall phase gates that hinder flow.

The principles and practices that enable the uninterrupted flow of value in SAFe are integral to the [Lean-Agile Mindset](#), [Value Stream Management](#), and Lean Thinking [1], which can be summarized as:

- Precisely specify value by product
- Identify the value stream for each product
- Make value flow without interruptions***
- Let the customer pull value from the producer
- Pursue perfection

This article, SAFe Principle 6, describes how to ***make value flow without interruptions*** by introducing Ÿighth flow accelerators that can be used to improve flow at any level of SAFe.

What is Flow?

First, it's essential to understand what SAFe means by *flow*. Flow *occurs when there is a smooth, linear, and fast movement of work product from step to step in a relevant value stream*. While the details of any flow system are based on its context, all flow systems have eight common properties, as illustrated in Figure 1.

Figure 1. Eight properties of a flow system

Figure 1. Eight properties of a flow system

Each is described briefly below:

1. **Work in process.** There is always some work in process in the system; if there weren't, there could be no flow of value.
2. **Bottlenecks.** In every flow system, one or more bottlenecks effectively limit the flow through the entire system.
3. **Handoffs.** Handoffs wouldn't be necessary if one person could do all the work. But in any material flow system, different individuals and teams will have different skills and responsibilities. Each plays its part in moving a work item through the system.

4. **Feedback.** Customer and stakeholder feedback is integral to efficient and effective outcomes. Ideally, feedback happens throughout the entire process.
5. **Batch.** As any system has a finite capacity, all the work can't be done at once. Therefore, work through the system occurs in batches designed to be as efficient as possible.
6. **Queue.** It all starts with a set of work items to be done. In addition, each value stream needs a prioritizing mechanism to sequence the work for the best value.
7. **Worker.** People do the critical work of moving work items from one state to another.
8. **Policies.** Policies are integral to flow. They may be local policies — like team-based policies that determine how a work item moves from step to step — or global policies like those that govern how work is performed within the company.

The Eight Flow Accelerators

Each flow property is subject to optimizations, and often many steps encounter unnecessary delays, bottlenecks, and other impediments to flow. *Making value flow without interruptions* can best be achieved by adopting the eight flow accelerators described in this article. These powerful accelerators of value are relevant to all Framework levels, but the challenges differ for each. An individual SAFe article discusses how these accelerators apply to each flow domain: [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#).

#1 Visualize and Limit WIP

Overloading teams and ARTs with more work than can be reasonably accomplished is a common and pernicious problem. Too much work in process (WIP) confuses priorities, causes frequent context switching, and increases overhead. It overloads people, scatters focus on immediate tasks, reduces productivity and throughput, and increases wait times for new functionality. Like a highway at rush hour, there is simply no upside to having more work in a system than the system can handle.

The first corrective action is to make the current WIP visible to all stakeholders. Figure 2 shows a simple Kanban board that illustrates the total amount of WIP and the process state of each work item. This Kanban serves as an initial process diagnostic, showing the current bottlenecks. Often, simply visualizing the current volume of work is the wake-up call that causes the organization to address the systemic problems of too much work and too little flow.

Figure 2. Kanban boards make excessive work-in-process (WIP) visible

Figure 2. Kanban boards make excessive work-in-process (WIP) visible

The following action is balancing the amount of WIP against the available development capacity. This is done by establishing and continually adjusting WIP limits for the relevant states. No new work is started when any workflow state reaches its WIP limit. This matches demand to capacity and increases flow through the system.

Limiting WIP, however, requires knowledge, discipline, and commitment. It may even seem counterintuitive to those who believe that the more work you put into the system, the more you get out. That can be true up to a point, but when the system becomes overloaded, throughput *decreases* dramatically. Indeed, there is no substitute for effectively managing WIP.

#2 Address Bottlenecks

Bottlenecks occur wherever people or resources (systems, materials, and so on) in the flow of value experience demand greater than the available capacity. Examples include a shortage of a specialized skill (such as a data scientist), insufficient

processing power for the build servers in the CI/CD pipeline, or a silicon supply shortage for building the integrated circuits of a cyber-physical system. Work piles up at a bottleneck and limits the effective throughput of value, as Figure 3 illustrates.

Figure 3. Bottlenecks reduce the flow of value through the value stream

Figure 3. Bottlenecks reduce the flow of value through the value stream

Upstream processes are blocked from moving value. Downstream processes are starved and waiting. Bottlenecks cause the value stream to operate slowly and uneconomically, far below its potential capacity. This is emphasized in the Theory of Constraints (Goldratt [2], [3]), which posits that the throughput of any flow system is limited by the capacity of a dominant constraint, or *bottleneck*. By this theory, investment in optimizing the system at any other point than the dominant constraint is waste, as it will not improve the throughput of the system.

No matter the theory, bottlenecks must be addressed, by adding additional skills, people, or other resources at the bottleneck step. While that is not always easy to do, (there are reasons why the bottleneck is a bottleneck), eliminating dominant

bottlenecks must be a primary focus as throughput will not increase until the bottleneck is addressed.

In the meantime, however, the complex workflows in solution development provide other options. It's often the case that not all the work is single-threaded (must pass through the bottleneck step). There is often other work (different features, etc.) that does not have to pass through the bottleneck. In this case, teams can selectively pick and deliver other valuable work, thus increasing value throughput, while the dominant bottleneck is being addressed. In addition, relentless improvement drives us to improve wherever we can do so, and knowledge workers take pride in improving the processes that are under their control. In the end, the efficiency of every step matters.

But the fact is that bottlenecks limit throughput and negatively affect economic outcomes.

#3 Minimize Handoffs and Dependencies

Handoffs occur whenever there is a separation between knowledge, responsibility, action, and feedback [4]. For example, dependencies happen between teams when the work of one team cannot continue until related work by another team is completed (see Figure 4). Both result in development waste in the form of wait states in the flow of value. They can also lead to rework as the knowledge transfer is likely imperfect, causing further delays.

Figure 4. Excessive handoffs and dependencies made visible on the ART planning board

The best solution to overcome handoffs and dependencies is to create teams and ARTs with all the knowledge, resources, skills, and decision-making authority to create an end-to-end flow of value. However, unhealthy dependencies and handoffs can still occur even when teams and trains have all the skills to deliver end-to-end value. This can happen when there is a series of handoffs between Agile Team members or when there are delays while waiting on decisions to be made outside the ART. Activities like value stream mapping, retros, and the I&A problem-solving workshop can help identify the root causes and potential solutions.

#4 Get Faster Feedback

Learning is the foundation of improvement and the engine that powers product development [4]. Doing this as fast as possible speeds up and improves the overall development process. The goal is to get positive and negative feedback into the

development process as early as possible.

However, we often discover that getting early feedback can be difficult for a variety of reasons, for example:

- Lack of direct access to customers
- Delays in the development value stream
- Late or infrequent integration results in discovering hidden work and defects
- Developing more functionality than what's needed
- Building the wrong things or more functionality than what's needed

Fast feedback is generally achieved by applying the basic Plan-Do-Check-Adjust (PDCA) learning cycle. However, to accelerate flow further, we've found that more needs to be done, for example:

- Applying customer centricity and design thinking as part of product development and engaging with customers frequently
- Making improvements to the continuous delivery pipeline, including build and test automation, test-first practices, and continuous integration
- Keeping work items small results in faster working increments of value.
- Using built-in quality practices, mob work, pairing, and swarming to increase team cohesion and focus on finishing one backlog item at a time
- Upholding a solid Definition of Done (DoD) to help teams work together to finish Increments of value and share knowledge
- Use "stop-the-line" to fix problems when they occur so they don't pile up.

Generally, solution builders need two types of feedback from each PDCA cycle (Figure 5):

Figure 5: Every PDCA cycle collects two kinds of feedback

Figure 5: Every PDCA cycle collects two kinds of feedback

1. **Feedback about building the right thing.** This feedback can only come from those users, customers, and economic stakeholders who can measure a solution's actual value. Each PDCA cycle is an opportunity for this learning, from early mockups and storyboards to system demos during development to feedback on pre-releases and deployed systems in production.
2. **Feedback about building it right.** Innovative systems constantly push the bounds of technology and the developers' skills. Each PDCA cycle also evaluates if the right technology is applied to optimally solve the customer's problem and meet the critical nonfunctional requirements (system qualities) that characterize robust and effective solutions.

Creating the mechanisms and processes to collect a wide range of data is the critical first step to promoting flow with faster feedback, but it doesn't stop there. The information should be quickly analyzed and evaluated to make effective adjustments and initiate the next PDCA cycle based on these learnings.

#5 Work in Smaller Batches

Faster feedback is one of the primary reasons for working in smaller batches. The smaller the size, the faster teams can collect and evaluate the feedback to adjust. In addition, smaller batches reduce WIP by limiting the number of requirements, designs, code, tests, and other work items moving through the system at any point. Smaller batches go through the system faster and with less variability, fostering faster learning. Moreover, since each item in the batch has some variability, larger batches accumulate more variability.

The economically optimal batch size depends on the holding cost (the cost for delayed feedback, inventory decay, delayed value delivery, and so on) and the transaction cost (the cost of preparing and implementing the batch). Figure 6 illustrates the tradeoff u-curve optimization for batch size [5].

Figure 6. U-curve optimization for batch size showing tradeoffs between holding cost and transaction cost

To improve the economics of processing smaller batches—teams should focus on reducing the transaction costs—resulting in higher throughput for a batch of a

given size. *Reducing batch size typically involves investment in automating the Continuous Delivery Pipeline*, including infrastructure and automation, continuous integration, builds, regression testing, and more. Shorter iterations and PIs also help to reduce batch size.

#6 Reduce Queue Length

Reducing the length of the queue that is feeding the system is another critical way to accelerate flow. As we have all experienced, long queues are fundamentally bad. They introduce waste, delays, and information decay. In addition, Little's Law (Figure 7) informs us that the *average wait time* equals the *average queue length* divided by the *average processing rate*. (While this might sound complicated, even the line at Starbucks illustrates that.) Therefore, assuming any average processing rate, the longer the queue, the longer the wait.

Figure 7. Long queues slow service and flow; Little's Law predicts the average wait time

For solution development, the longer the queue of committed work awaiting implementation, the longer the wait time for new features, regardless of the team's efficiency. For example, suppose an ART has an average flow velocity of 10 features per quarter and a committed backlog of 30. In that case, the customer may have to wait as long as three quarters before any new features can start developing. This example explains why queues are fundamentally bad and can significantly delay the ability to respond to customer needs.

Reducing queue length decreases delays, reduces waste, increases flow, and improves predictability. It's a requisite for faster service and a more consistent flow of value.

#7 Optimize Time \hat{O} in the Zone \tilde{O}

Being \hat{O} in the zone \tilde{O} (also described as being in a \hat{O} flow state \tilde{O}) is an engaged mental state of extreme focus on an activity where the work feels effortless and time passes quickly. People and teams in the zone demonstrate higher creativity, productivity, happiness, and fulfillment. Getting into this mental state requires uninterrupted focus time, autonomy, competence, and connectedness to others to engender self-actualization and intrinsic motivation. [5]

Contrast this to the conditions in a typical work environment where work occurs in functional siloes in a batch-queue-handoff system. These frequent interruptions (emergency requests, ad hoc status reports, constant communication alerts, and so on) are the norm, and excessive WIP drives frequent task switching. (Figure 8).

Figure 8. Time' in the zone' is often a fraction of the total workday

Figure 8. Time in the zone is often a fraction of the total workday

There is an essential connection between creating a continuous flow of value and creating a working environment where individuals and teams can maximize their time in the zone. Knowledge workers also need the time and space free from interruption essential for complex tasks involving application, analysis, evaluation, and creativity, and ultimately the personal satisfaction that completion engenders.

#8 Remediate Legacy Policies and Practices

During or after a Lean-Agile transformation, enterprises must constantly look out for legacy policies and practices that inhibit flow (Figure 9). Many of these practices became part of the culture and are described as ‘we’ve always done it this way,’ even when they are no longer fit for purpose. Examples are many, some legendary:

Continued reliance on phase-gate milestones and the iron triangle of fixed scope, resources, and time. In reality, all three components become fixed instead of tradeoffs among constraints.

Obsolete or unnecessary change control boards, including extraneous oversight and

reporting

Waterfall-based quality management systems for regulations and compliance

Obsolete tech standards—design specifications, audit practices, and the like—in environments where they are not mandated or required for quality

Continuation of timesheet reporting *in addition to* Agile Lifecycle Management (ALM) tooling, requiring double recording of time

Traditional HR performance reviews and compensation policies that cause unhealthy competition

Agile is adopted only by teams; the mindset of management and portfolio governance remain unchanged

Figure 9. The maze of legacy policies and practices

Figure 9. The maze of legacy policies and practices

And that's not an exhaustive list. While many of these patterns may well have solved problems in the past, they now create *new* problems that become ongoing impediments to flow. They must be proactively or reactively discovered, eliminated, modified, or mitigated.

Measuring Flow

It's difficult to improve what isn't measured. And historically, it has been challenging to measure the development process as the work items are mostly intangible and invisible, living mainly in the minds of the knowledge workers who design and build the systems. But the physics and tooling of flow (timeboxes, Kanban, value stream mapping, CD pipeline, stories, features, and more) provide a new basis for measuring the flow of value through the development value stream. The SAFe [Measure and Grow](#) article describes six metrics —flow *distribution, velocity, time, load, efficiency, and predictability*—for measuring flow (see Figure 10).

Figure 10. SAFe flow metrics

Figure 10. SAFe flow metrics

Flow Distribution is a measure of the proportion of work items by type in a system.

Flow Velocity measures the number of completed work items over a time period.

Flow Time is a measure of the time elapsed from start to completion for a given work item.

Flow Load is a measure of the number of work items currently in progress (active or waiting).

Flow Efficiency is the ratio of the total time spent in value-added work activities divided by the total flow time.

Flow Predictability is a measure of how consistently teams, ARTs, and portfolios are able to meet their commitments.

Together, these metrics provide a comprehensive view as new value flows through the development value stream. These measures need to be relatively easy to collect, maintain, and visible to be valuable and actionable. Fortunately, they can be automated using most modern Agile Lifecycle Management (ALM) tooling. In addition, organizations should complement the flow metrics with qualitative data to ensure their delivery flow creates the right solutions for customers when needed.

Summary

These eight flow accelerators help teams increase throughput and deliver value faster. As an added benefit, implementing them gives people a sense of control over the process and triggers fast and measurable improvements in customer satisfaction and employee engagement.

Learn More

[1] Womack, James P., and Jones, Daniel T. *Lean Thinking: Banish Waste and Create Wealth in Your Organization*. Free Press, 2003.

[2] Goldratt, Eliyahu M. *The Goal: A Process of Ongoing Improvement*. The North River Press Publishing Corporation, 1986

[3] Goldratt, E. M. *What is this Thing called Theory of Constraints and How should it be Implemented?* North River Press, Inc, 1990

[4] Oosterwal, Dantar P. *The Lean Machine*. AMACOM, 2010

[5] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas, 2009.

[6] Ward, Allen, and Durward Sobeck. *Lean Product and Process Development*. Lean Enterprise Institute, 2014.

[7] Csikszentmihalyi, Mihaly. *Flow*. HarperCollins, 1990

[8] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. IT Revolution Press, 2018.

Last update: 6 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Delay is the enemy
of progress*

— Eliot Spitzer

Solution Train Flow

Solution Train Flow describes a state where a Solution Train delivers a continuous flow of valuable capabilities to the customer.

The SAFe Enterprise Solution Delivery competency guides how ARTs can work together with an aligned mission to build some of the world's largest and most critical systems. This competency has proven effective in improving business outcomes for SAFe Enterprises building these systems.

Adjust Cookie Settings

But solution delivery at scale is complex and requires unprecedented cooperation between release trains, suppliers, customers, and stakeholders. The world relies on these substantial systems, and every delay is a lost opportunity to help those depending on them. This article describes how to provide a continuous flow of value delivery, even when building the world's largest and most complex systems.

Note: About the Flow Article Series

SAFe is a flow-based system. As such, any interruptions to flow must be identified and addressed systematically to enable continuous value delivery. While flow-based guidance is embedded throughout SAFe, a special collection of six articles directly addresses impediments to flow. These are [Value Stream Management](#), [Principle #6- Make value flow without interruptions](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), [Portfolio Flow](#), and summarized in the extended Guidance article [Accelerating Flow with SAFe](#). These articles define flow along with a set of eight flow accelerators that teams can use to address, optimize, and debug issues with achieving continuous flow. This article describes how to make value flow without interruption.

Details

SAFe defines a set of eight flow accelerators that can address, optimize, and debug issues to achieve continuous flow at any level of the Framework. This article describes how Solution Trains can apply these eight accelerators to improve the flow of value for large and complex systems. Each accelerator is described in the sections below.

#1 Visualize and Limit WIP

Why it matters

Building big systems requires a lot of work. In response, it is common that the development process is overloaded with more work than can be reasonably accomplished in the time allotted. Excessive Work in Process (WIP) overburdens development leading to burnout, missed deadlines, cost overruns, and ultimately,

poor economic outcomes.

What to do about it

Fortunately, managing WIP is a recognized and addressable problem. Common techniques include:

Make all WIP visible. Solution Train WIP comes from two sources (Figure 1). The Solution Train Kanban system identifies and tracks new solution Capabilities implementation. In addition, the ARTs have work that arises locally to support their solution's vision. Both sources add to the work in the development system. Seeing, understanding, and respecting *both sources* is the key to evaluating the entire system's work volume and limiting it to improve throughput.

Figure 1. Visualize and reduce WIP by recognizing its sources

Figure 1. Visualize and reduce WIP by recognizing two primary sources

Align work to available capacity. Solution Train leaders should understand their ART's and suppliers' capacities to ensure system throughput. The Solution Roadmap must typically be continually adjusted to align future work (new Epics and

Capabilities) to the available capacity.

#2 Address Bottlenecks

Why it matters

It takes significant resources to build large solutions—people, supply chains, organizations, existing systems, and materials. To achieve the intended results, they all must work in concert. Bottlenecks must be identified and addressed continuously.

What to do about it

Discover and address bottlenecks. There are many ways to uncover bottlenecks:

- Visualizing the WIP in the stages of the development pipeline
- Observing and surfacing the pain of delays and missed milestones
- Listening to development teams’ feedback about stresses and problems in the system

Events such as [Inspect & Adapt](#) also help proactively identify and address bottlenecks. However they are discovered, there are multiple ways to remediate them:

Increase the capacity at the bottleneck. Increasing capacity at bottlenecks is an obvious first step. Address bottlenecks caused by insufficient knowledge or skills by employing additional teams or [Shared Services](#), leveraging other suppliers, or creating more “T-shaped” ARTs and teams to flex to the bottleneck activities (see [Enterprise Solution Delivery](#)). Bottlenecks caused by insufficient resources (lack of testing environments) or inadequate automation (manual build and integration steps) require proper investment to mitigate them.

Move work around the bottleneck. Where bottlenecks persist, adjust the Solution Roadmap to focus on delivering value that does not have to work through bottleneck resources [1].

Move work between suppliers. Ensure that contracts and internal governance rules permit teams to move the work freely to those with capacity. For example, confirm that governance rules allow everyone reasonable access to development assets (codebases) and that contracts do not put restrictions on adjusting the teams’ work (see [Agile Contracts](#)). Also, proactively create learning to enable more teams to flex towards critical types of work.

Decentralize decision-making. As teams and ARTs run faster, slow decision-making creates its own set of bottlenecks. For example, traditional leaders who have not embraced a [Lean-Agile Mindset](#) may struggle with delegating authority. Helping them understand the new way of defining and evaluating work leads to better, faster, and more decentralized decision-making.

#3 Minimize Handoffs and Dependencies

Why it matters

Large systems have many interconnections, making significant dependencies between systems builders common. However, excessive dependencies and handoffs disrupt the flow and create unnecessary context switching and overhead. Delays and cost overruns are typical results.

What to do about it

Although teams cannot eliminate handoffs and dependencies entirely, there are several ways to reduce them at scale, as described below.

Architect for scale and modularity. Architect large solutions to allow ARTs and teams to iterate independently over designs and release new functionality for feedback with minimal external collaboration.

Organize around a clearly defined context. Organize ARTs and teams around clear business and technical domains (see [Domain Modeling](#)), providing relatively stable organizational structures with fewer dependencies. Apply [Team and ART topologies](#) to support this and reduce organizational dependencies.

Create truly cross-functional ARTs and teams. To create an end-to-end flow of value, ensure every team and ART has sufficient knowledge, resources, skills, and decision-making authority to deliver value with minimal dependencies.

Organize to support innovation. Since change occurs more frequently in areas of high innovation, organize teams and ARTs around these areas early in the process. As the design emerges, apply the architectural and organizational practices (topologies) described above to decouple ARTs and teams.

Adopt Lean-Agile practices across the enterprise. Other parts of the organization—such as Procurement, IT, Release Management, and Configuration Management—traditionally work as independent functional departments. This structure, however, can create a silo effect, where work intake procedures do not align with new value delivery opportunities. Instead, introduce the whole organization to the Lean-Agile way of working and include them in ARTs where appropriate.

#4 Get Fast Feedback

Why it matters

Large solutions often have an unacceptably high cost of failure. Lack of proper feedback during development can lead to massive rework, unanticipated delays, unsatisfied customers, and unacceptable social and economic consequences. However, getting early feedback presents unique challenges for large solution

developers. Customers may work in other development value streams or are end-users far removed from the development process. To get their feedback quickly, teams must proactively identify their key customers—end users, other development teams, production, operations, etc.—across an end-to-end delivery pipeline. (Figure 2).

Figure 2. Enable fast feedback across the Solution Train

What to do about it

Find shortcuts to the customer where necessary. Early on, teams must determine who can best provide feedback and proactively break through organizational and other barriers to interact with those customers (Figure 3). Then, by defining personas and journey maps, teams can apply a customer-centric approach to external and internal stakeholders (see [Design Thinking](#)). For customers shared across teams and ARTs, create a common set of customer personas and journey maps.

Figure 3. Find shortcuts to internal and external customers

Figure 3. Find shortcuts to internal and external customers

Ensure capacity for feedback. Getting fast feedback is possible only when others must have the ability to provide it. Confirm that the necessary stakeholders understand the importance of their input and will allocate time to provide it. Likewise, system developers must be able to respond and adjust to the feedback they get.

Integrate frequently. Getting fast feedback requires the ability to integrate small changes into the larger system quickly. ARTs and teams invest in automating an end-to-end CDP that reduces the time and effort needed to deliver changes for feedback (see Figure 2). And where end-to-end integration is impractical, mockups and digital models help elicit early feedback.

Build quality in. When developing big systems, minor quality issues within components or subsystems can add up and become major problems for the complete system. Worse, they can be complicated to locate and diagnose, as there are possibly many contributing factors, including the simple accumulation of what would otherwise be minor glitches. To address this, ensure that everyone understands the importance of [Built-in Quality](#) and provides adequate testing environments for the [Solution Œ Contexts](#) at all levels (Figure 4).

Figure 4. Ensure quality and sufficient testing in a realistic context at every level

Figure 4. Ensure quality and sufficient testing in a realistic context at every level

#5 Work in Smaller Batches

Why it matters

Traditionally, large system builders first specify requirements, design and build the system, test it, and release large batches of new functionality. This restricts flow and delays feedback, resulting in added rework, late delivery, and unsatisfied customers.

What to do about it

Specify, design, and implement the solution with small, vertically-sliced value increments. Implementing in small batches requires Solution Train leaders to specify and design the system incrementally. Use PI timeboxes to scope new capabilities and exploration activities (See *Continually Refine the Fixed/Variable Solution Intent* in [Enterprise Solution Delivery](#)).

Focus on critical knowledge creation. To reduce batch sizes, focus Teams and ARTs on what they need to learn from the technical, market, or user-acceptance perspectives. Then converge on the work necessary to gain that knowledge (see [SAFe Principle #4, Build incrementally with fast learning cycles](#)).

Apply a standard cadence across the Solution Train. A shared cadence is a forcing function that helps reduce batch sizes for all contributors. Each PI becomes a Ôpull eventÕ that gathers small changes to solution elements into an integrated whole for review and feedback.

Optimize the release batch. Traditionally, large systems are built using a single, significant, and essentially one-time development effort. Conversely, an Agile approach builds the system incrementally, enabling early and frequent releases of value (see [Enterprise Solution Delivery](#)). Releasing frequently adds pressure to optimize the release batch. Institute a fast, reliable release process that ensures compliance, allows limited releases to target specific customer segments, and employs a rollback strategy to avoid service disruption when errors occur (see [Release on Demand](#)).

#6 Reduce Queue Lengths

Why it matters

Traditional approaches to specifying, designing, and implementing big systems often result in fixed, multi-year schedules, culminating in long, committed queues of unfinished work. These long queues increase wait times for new functionality and inhibit responsiveness to change. While some degree of long-term forecasting and fixed, significant milestones will likely remain, these implied queues of Ôbig things yet to doÕ must be managed to improve flow time.

What to do about it

Replace fixed schedules with rolling-wave roadmaps. Wherever possible, replacing fixed plans with connected, rolling-wave roadmaps adds flexibility and agility to system design and delivery. The Solution TrainÕs roadmap communicates the high-level, rolling-wave forecast of milestones, dates, and Epics that drive connected PI roadmaps for ARTs and suppliers. Roadmaps must be adjusted frequently as new technologies, users, and business facts emerge (Figure 5).

Figure 5. Forecast and adjust work with connected roadmaps

Figure 5. Forecast and adjust work with connected roadmaps

Apply a set-based design to fixed milestones. Due to the unacceptably high cost of change and the need for coordination (flight test, significant product release, and so on), some milestone dates will likely be fixed and unchangeable. In these situations, apply [Set-Based Design](#) to ensure sufficient flexibility in meeting these objectives.

#7 Optimize Time ÐIn the ZoneÐ

Why it matters

Large systems have significant scope, complexity, and governance requirements. This creates an environment where ARTs and teams often spend too much time multiplexing, context switching, and reporting Ð and too little time building the solution. Optimizing time Ðin the zoneÐ allows practitioners to focus on solution-building without interruption.

What to do about it

Apply sound system and organizational design. The activities of building large systems can cause teams to spend too much time blocked by technical or business concerns outside their skills, purpose, and mission. Instead, design the system and organizational structures with good separation of concerns, minimize dependencies, and reduce meeting time. This frees the teams to focus on building their part of the work.

Ensure effective SAFe events. Ensure ART and Solution Train PI Planning and sync events are productive. Reduce or eliminate extraneous and redundant meetings, and manual status reporting activities.

#8 Remediate Legacy Practices and Policies

Why it matters

As a result of their scope, costs, and expected impact, large solution development activities are highly visible and attract significant governance and oversight. Many of these oversight activities are legacy practices that addressed past problems but now create new problems that impede flow. They must be discovered and then either eliminated, modified, or mitigated.

What to watch out for

The examples are many; some are legendary:

Traditional project and program management models—such as earned value management (EVM), work breakdown structure (WBS), and project-based cost accounting—are placed *on top of* Agile development

Quality and governance organizations that force waterfall, stage-gated milestones
Legacy supplier management and contracts that define fixed scope, schedule, and deliverables

Incentivizes that encourage people to optimize building *their parts* of the system versus focusing on the overall system

Lack of transparency with and across suppliers hiding critical facts until it's too late to address them

ÓColor of money,Ó politics, or (perceived) legal concerns that prevent the appropriate training and support for suppliers

Restricted access to the end customers needed to validate the solution

Engaging verification and validation (V&V) and compliance groups only at the end

What to do about it

Change agents must constantly watch for and recognize these legacy governance rules and practices. In particular, stakeholders in compliance and regulatory functions—such as quality, certification, and release management—may view a

Lean-Agile transformation as a risk to their proven approach. Help them obtain the knowledge and information they need to operate successfully in a SAFe development environment by engaging them early and often.

Measure and Improve Flow

And finally, it's difficult to improve what isn't measured. Flow is one of SAFe's three measurement domains (see [Measure and Grow](#)) and helps Solution Trains improve by revealing issues and areas for improvement. While all SAFe's Flow Metrics apply well to the Solution Train, flow *time*, *load*, and *distribution* are particularly relevant and are discussed below.

Flow Time

Building big systems creates a lot of activity, with hundreds or thousands of individuals busy working on their parts of the system. Solution Train leaders must ensure their collective work produces overall value towards a common goal. Flow time measures the interval for solution-level Capabilities and Epics to move through all workflow steps in the Kanban. It shows the ARTs' and teams' ability to deliver work against common objectives and is a key indicator that supports planning activities.

The ARTs on a Solution Train have independent work, which ultimately contributes to the overall value of the Solution Train. To ensure all parts of the system are functioning well, Solution Train leaders can track both ART and Solution Train flow time (Figure 6).

Figure 6. To understand Solution Train Flow Time, include the ART's Flow Times

Figure 6. Solution Train flow time aggregates the ART's flow times

Flow Load

As we noted earlier, the need to meet deadlines encourages management to put more work into the system. But this excessive load increasesÑrather than decreasesÑflow time. Instead, Solution Train leaders should focus first on completing existing work. One way to understand the load over time is to measure and analyze cumulative flow with a cumulative flow diagram (CFD). This gives leaders insights into how to manage the flow of work better. (Figure 7).

Figure 7. Cumulative flow diagram

Figure 7. The cumulative flow diagram shows the work in the system at various stages over time

Flow Distribution

Builders of large systems often cannot rely on open-source and out-of-the-box solutions for infrastructure. Even when they can, they must develop even more on their own. Moreover, significant risks abound, and these must be addressed as part of the work backlog. This means that Solution Train leaders must prioritize and balance enabler and risk-mitigation activities with other types of work. Figure 8 shows a flow distribution spanning three types of work items—features, enablers, and maintenance—over a period. Leaders use this information to ensure a healthy balance of the types of work in the system.

Figure 8. The types of work in the system must be balanced over time

Figure 8. The types of work in the system must be balanced over time

Last Updated: 9 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



While you may ignore economics, it won't ignore you.

—Don Reinertsen,
Principles of Product Development Flow

Principle #1 → Take an economic view

[Adjust Cookie Settings](#)

Realizing the goal of Lean—achieving the shortest sustainable lead time with the best quality and value—requires understanding the economics of a mission. Without that, even a technically capable system may cost too much to develop, take too long to deliver, or incur excessive manufacturing or operating costs.

That’s why the entire chain of leadership, management, and knowledge workers must understand the economic impact of their choices. Traditionally, the budgets for their activities were known only to the decision-makers who understood the business, marketplace, and customer finances. This meant that a worker’s everyday decisions were made without this information or escalated to those who had it. That resulted in delays, inefficiencies, and a lack of empowerment by the people closest to the work.

Therefore, SAFe’s first Lean-Agile Principle is to *take an economic view*. It is Principle #1 for a reason. Economics should inform and drive decisions at all levels, from Portfolio to Agile Teams. While many factors can contribute to failed solutions, poor economics is one of the most common.

This article describes the two practices essential to achieving optimum economic outcomes via Lean-Agile methods:

- Deliver early and often
- Apply an economic framework

Each is outlined in the sections below. And although every knowledge worker needs to understand them, many of these principles are already embedded directly into various SAFe practices.

Deliver Early and Often

[Enterprises](#) decide to embrace Lean-Agile development either because their existing processes aren’t producing the desired results or because they anticipate that they won’t work in the future. By choosing a Lean-Agile path, they’re embracing a model based on incremental development and early and continuous value delivery, as Figure 1 illustrates.

Figure 1. Moving to early and continuous delivery

Figure 1. Moving to early and continuous delivery

The ability to deliver early and often has a direct economic benefit, as illustrated in Figure 2.

Figure 2. Incremental development and delivery produce value far earlier

Figure 2. Incremental development and delivery produce value far earlier

This figure shows how Lean-Agile methods deliver value to the customer much earlier in the process. What's more, this value accumulates over time: The earlier and longer the customer has it, the more value they receive. Conversely, with the waterfall model, value can't even begin to accumulate until the end of the development cycle.

This difference is a material economic benefit of SAFe. In addition, the figure above doesn't account for the advantage of far faster feedback or the probability that the waterfall delivery would not occur on time. And there is a third factor, as shown in Figure 3.



Figure 3. Value is higher early on, producing higher margins over a longer period

Figure 3 illustrates another key differentiator. As long as the quality is high enough, products and services delivered to market early are typically more valuable. After all, if they arrive ahead of the competition, they are worth a premium as they aren't available from anyone else. Over time, features become commoditized. Cost, not value differentiation, then rules the day. Even a Minimum Viable Product (MVP) can be worth more to an early buyer than a more full-featured product delivered later.

The net effect is that cumulative gross margins are higher. For government agencies and non-profits where gross margin is not a factor, the same concept results in more effective value delivery for the funds provided by taxpayers and donors. This is the basic premise of Lean-Agile development, firmly entrenched in the [Lean-Agile Mindset](#).

Apply a Comprehensive Economic Framework

But as important as it is, delivering early and often isn't the entire picture.

Additionally, every SAFe portfolio requires an *Economic Framework*—a set of decision guidelines that align everyone with the financial objectives of a Portfolio and inform the continuous decision-making process. After all, teams and Agile Release Trains (ARTs) make small and large decisions every day. Their choices can influence economic outcomes positively or negatively. Without guidance, self-organizing teams will make their “best guess.” Although this can result in decentralized decisions, they may not be aligned with the core economics of the system, creating the potential for technical debt, rework, waste, and a lack of fitness for use.

As illustrated in Figure 4, SAFe’s economic framework contains four primary elements:

- Operating within lean budgets and guardrails
- Understanding solution economic trade-offs
- Leveraging suppliers
- Sequencing jobs for the maximum benefit

Figure 4. The four primary elements of the economic framework

Figure 4. The four primary aspects of the economic framework

Operate Within Lean Budgets and Guardrails

As described in the [Lean Budgets](#) article, one of the most significant transitions for the Lean-Agile enterprise is to move from project-based, cost-center accounting to a more streamlined budget process. The new model allocates the funding to long-lived portfolio value streams. Budget *guardrails* (Figure 5) inform the oversight and governance that guide ongoing spending decisions to support these higher-level budgets.

Figure 5. Lean budgets and guardrails govern value stream spend

Figure 5. Lean budgets and guardrails govern value stream spend

As described in the [Guardrails](#) article, SAFe suggests four specific guardrails:

- Guide investments by horizon
- Optimize value and solution integrity with capacity allocation
- Approve significant initiatives
- Continuous business owner engagement

Every solution inherits the guardrails of the corresponding value stream as defined by [Lean Portfolio Management](#) (LPM). Budgets and guardrails define the boundaries within which Solution Train and ART leaders make decisions on the definition, scope, and sequence of epics, capabilities, and features. As context changes, these guidelines are updated periodically.

Understand Solution Economic Trade-Offs

Within the degrees of freedom that lean budgets and guardrails provide, there are still many alternatives and trade-offs for what to build and how and when to build it. Reinertsen [1] identifies five considerations that affect these decisions:

- Development expense** ☰ the cost of labor and materials required to implement a capability
- Lead time** ☰ the time needed to implement the capability (described as “Cycle time” in Reinertsen’s work)
- Product cost** ☰ the manufacturing cost (of goods sold) and deployment and operational costs
- Value** ☰ the economic worth of the capability to the business and the customer
- Risk** ☰ the uncertainty of the solution’s technical or business success

Figure 6. Multiple variables influence economic decisions

The arrows in the diagram in Figure 6 are the key to the model. They illustrate that changing any variable can impact one or more of the others. For example, a company can add developers to an initiative that's behind schedule, increasing the development cost. And if development costs drive the product costs up, economical pricing may become unrealistic. Understanding how each variable influences the others is vital to making the proper trade-offs.

Leverage Suppliers



End the practice of awarding business on the basis of price tag. Instead, minimize total cost. Move toward a single supplier for any one item, on a long-term relationship of loyalty and trust.

— W. Edwards Deming

Figure 7. Large solutions typically depend on one or more external suppliers

Figure 7. Large solutions typically depend on one or more external suppliers

To augment their internal workforce or to provide components within the system-of-systems, most organizations building large solutions depend on external suppliers. They do this for two reasons:

1. **Insufficient workforce capacity.** Outsourcing can provide a cost-efficient way to add personnel, especially if the need is temporary or when demand is highly variable.
2. **Availability of specialty components, personnel, or skill sets.** A supplier may provide specific hardware, software, or skills needed for the solution. It may be significantly more economical to buy and integrate the component than build it.

Since the money directed to external suppliers can represent a significant percentage of the overall cost of the solution, decisions regarding suppliers are a critical component of the economic framework.

An essential part of supplier economics is the relationship between the buyer and the supplier:

Some relationships are transactional. Competitive pricing can create favorable economics in a transactional relationship, such as purchasing off-the-shelf parts or components from a third party or routine staff augmentation.

Other relationships are true partnerships, persistent and long-lived. Often, the buyer and supplier are co-developing the solution. In this instance, economic considerations are driven by longer-term concerns. These may include residual licensing, implicit knowledge, ownership of intellectual property, or other considerations that recognize the dependence on the specialty knowledge the supplier brings to the table.

In either case, as Deming notes, the most favorable economics emerge from longer-term considerations, mutual trust, and relationships where the supplier and buyer each have favorable economics (for example, see [Agile Contracts](#) for more detail).

In other cases, the best economics can be achieved through mergers and acquisitions of companies with long-term strategic value to the solution and the market’s competitive landscape.

Read the Suppliers article for further guidance on how to incorporate suppliers into solution development.

Sequence Jobs for Maximum Benefit

Every significant initiative has many new backlog features and capabilities just waiting to be implemented. However, SAFe is a flow-based system in which job-sequencing optimizes economics versus theoretical job return on investment (or worse, first-come, first-served, loudest voice, or other historical patterns). Program and Solution Train Kanban systems and the ART and Solution Train backlogs enable this. To minimize the cost of delay (COD), jobs are pulled into implementation based on Weighted Shortest Job First (WSJF). WSJF prioritizes the backlog to ensure the highest value is delivered in the shortest lead time.

Figure 8. Weighted Shortest Job First (WSJF) is the recommended technique for economically sequencing the backlog.

Practices Provide the Form. People Make the Decisions

This article described some mechanisms for economic decision-making, the foundation for effective management based on economics. SAFe also defines the roles and responsibilities of people in the decision-making chain. However, those decisions don't make themselves. [Lean-Agile Leaders](#) with the relevant context, knowledge, and authority lead the process. Of course, they don't act alone. They empower practitioners and work with their larger stakeholder community to determine the best path. The concepts of the economic framework ensure that responsible decision-making happens throughout the development organization, bringing all the economic benefits of Lean-Agile development to the entire enterprise.

Learn More

[1] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

Last update: 31 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd, Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



A system must be managed. It will not manage itself. Left to themselves, components become selfish, competitive, independent profit centers, and thus destroy the system. The secret is cooperation between components toward the aim of the organization.

Adjust Cookie Settings

Principle #2 Ð Apply systems thinking

The four foundational bodies of knowledge that inform SAFe are systems thinking, Agile development, Lean product development, and DevOps. Systems thinking takes a holistic approach to solution development, incorporating all aspects of a system and its environment into the system's design, development, deployment, and maintenance.

Figure 1 illustrates three primary aspects of systems thinking.

Figure 1. Three aspects of systems thinking

Figure 1. Three aspects of systems thinking

Understanding these concepts helps leaders and teams navigate the complexity of solution development, the organization, and the larger picture of total time-to-market. Each is described in the following sections.

The Solution Is a System

SAFe guides the development and deployment of complex technology-based solutions. Solutions are represented by the SAFe Solution object, the tangible object that delivers the end user value and is the subject of each development value stream—the application, satellite, medical device, or website. When it comes to such tangible systems, Deming’s comment that “a system must be managed” leads to some critical insights:

Team members must clearly understand the system boundaries and how it interacts with the environment and the systems around it.

Optimizing a component of the system does not optimize the whole system.

Components can become selfish and hog the resources—computing power, memory, electrical power, whatever—that other elements need.

For the system to behave well, teams must understand the intended behavior and architecture (how the components work together to accomplish the system’s aim). Intentional design is fundamental to systems thinking.

The value of a system passes through its interconnections. Those interfaces—and the dependencies they create—are critical to providing ultimate value. Continuous attention to those interfaces and interactions is vital.

A system can evolve no faster than its slowest integration point. The faster the full system can be integrated and evaluated, the quicker the system knowledge grows.

The Enterprise Building the System Is a System, Too

There’s a second aspect to systems thinking: the people, management, and processes of the organization that builds the system are also a system. The understanding that “systems must be managed” applies here as well. Otherwise, the components of the organization building the system will optimize locally and become selfish, limiting the speed and quality of value delivery. This leads to another set of systems thinking insights about the enterprise:

Building complex systems is a social endeavor. Therefore, leaders must cultivate an environment where people collaborate on the best way to build better systems.

Suppliers and customers are integral to the development value stream. Both must be treated as partners based on a long-term foundation of trust.

Optimizing a component does not optimize the system in this case, either. Therefore optimizing local teams or functional departments does not enhance the flow of value through the enterprise.

And as with physical systems, the value of the system passes through its interfaces here too. Accelerating flow requires eliminating functional silos and creating cross-functional organizations, such as Agile Teams, Agile Release Trains (ARTs), and Solution Trains.

Understand and Optimize the Full Development Value Stream

Development value streams are fundamental to SAFe. A SAFe portfolio is a collection of development value streams, each delivering one or more solutions to the market. As illustrated in Figure 2, each development value stream consists of the steps necessary to integrate and deploy a new concept through a new or existing system.

Figure 2. The solution development Value Stream

Figure 2. The solution development Value Stream

Understanding and optimizing the entire development value stream—the third aspect of systems thinking—is the only way to reduce the total time it takes to go from concept to cash [2]. Systems thinking mandates that leaders and practitioners grasp and continuously optimize the entire development value stream, especially as it crosses technical and organizational boundaries.

One essential process is value stream mapping, a systematic way to view all the steps required to produce value. Value Stream Mapping (Figure 3) helps leaders quickly recognize that the actual value-added processing steps—creating code and components, deployment, validation, etc.—consume only a small portion of the total time-to-market. This recognition drives these leaders to constantly focus on the delays between steps.

Figure 3. Value Stream mapping example: most of the time is wait time

Figure 3. Value Stream mapping example: most of the time is wait time

Note that in this example that almost all the time between a feature request and deployment is wait time, resulting in a highly inefficient process.

Only Management Can Change the System

“Everyone is already doing their best; the problems are with the system. É only management can change the system.”

NW. Edwards Deming

This Deming quote prepares us for a final set of insights. Systems thinking requires a new approach to management as well, a perspective where managers are problem solvers, take the long view, proactively eliminate impediments, and lead the changes necessary to improve systems and performance. These [Lean-Agile Leaders](#):

Exhibit and teach systems thinking and Lean-Agile values, principles, and practices
Engage in solving problems and eliminating roadblocks and ineffective internal systems
Apply and teach root-cause analysis and corrective action techniques
Collaborate with the teams to reflect at key Milestones and identify and address shortcomings
Take a long-term view, investing in enabling capabilities such as infrastructure, practices, tools, and training that lead to faster value delivery, better quality, and higher productivity
Foster a [Continuous Learning Culture](#) that includes relentless improvement in the application of systems thinking

Summary

Understanding the elements of systems thinking helps leaders and teams recognize the ÔwhyÕ and the ÔwhatÕ of their actions, as well as the impact on those around them. This understanding leads to a leaner, smarter enterprise that can better navigate organization and solution development complexities. And that results in better business outcomes.

Learn More

[1] Deming, W. Edwards. *The New Economics*. MIT Press, 1994.

[2] Poppendieck, Mary, and Tom Poppendieck. *Implementing Lean Software Development*. Addison-Wesley, 2006.

Last update: 14 March 2023

FRAMEWORK

[Download SAFe
Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner
Partner Directory](#)

GET SOCIAL

[Twitter
Linkedin
YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks
Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy
Cookie Policy
Your California
Consumer Rights](#)

English

Scaled Agile Framework



Principle of Alignment: There is more value created with overall alignment than with local excellence.

—Don Reinertsen

Solution Train

The Solution Train is the organizational construct used to build large solutions that requires the coordination of multiple ARTs and suppliers.

The Solution Train construct in SAFe describes the additional roles, events, and artifacts needed to build large solutions that require contributions from multiple ARTs and suppliers. These solutions often have an unacceptable social or economic cost for failure, are often subject to industry and regulatory standards and must provide objective evidence of compliance with those standards. Many large solutions are part of an even larger Ÿsystem-of-systemsÕ built by an extended supply chain, as described in the [Supplier](#) article.

Details

Solution Trains provide the constructs businesses need to build large solutions— including cyber-physical systems—in a Lean-Agile manner. They align multiple [Agile Release Trains](#) (ARTs) and [Suppliers](#) to a shared mission (Figure 1). Solution Trains help manage the inherent risk and variability of large-scale solution development with additional roles, artifacts, and events described in this article.

Figure 1. ARTs and suppliers power the Solution Train

Figure 1. ARTs and suppliers power the Solution Train

Why Solution Trains?

Solution Trains form for different reasons. Some trains form with new ARTs and [Agile Teams](#) to specifically address a large initiative, while others begin by combining existing ARTs and teams. This is particularly true in organizations operating with SAFe for some time as they recognize the opportunity and need to reorganize around value (see Principle # 10) occasionally.

In the latter case, Solution Trains may experience an initial forming period even though its ARTs have been performing for quite some time with a mature backlog, roadmap, [Continuous Delivery Pipeline](#) (CDP) infrastructure, and other assets. When forming a Solution Train from existing ARTs, quickly creating the foundational SAFe artifacts (Solution Train [Vision](#), [Backlog](#), and roles) is critical to getting work to the existing ARTs and teams. Over time, the leaders can grow the backlog's Solution Train Backlog Kanban system, create a [Solution Roadmap](#), form a [System Team](#), and implement other large solution constructs.

Solution Train Characteristics

Agile Release Trains Power the Solution Train

Each ART within a Solution Train contributes to the development of the solution, as shown in Figure 2. All development activities typically occur within each ART and are coordinated by the Solution Train.

To support the overall goal of continuous value delivery to the customer, each ART within the Solution Train must be designed to maximize flow across the entire Solution Train.

As described in the [Agile Teams](#) and [Agile Release Train](#) articles, SAFe recognizes fundamental team topologies from [1] to help with the team and ART design. These topologies can be readily extended to help make the right trade-offs in ART design as part of a Solution Train, as shown in Figure 2 and described below.

Figure 2. A mixture of topologies applied to ARTs within a Solution Train

Figure 2. A mixture of topologies applied to ARTs within a Solution Train

Stream-aligned ARTs

A stream-aligned ART, just like a stream-aligned team, will have the necessary personnel, skills, and authority to deliver value, whether a whole product, service, or portion of the overall solution to the end user. For example, a navigation system for an autonomous vehicle may be developed and delivered by an integrated stream-aligned ART.

Complicated subsystem ART

Most large systems also include extensive subsystems. Complicated subsystem ARTs are common when building large-scale systems. These are unique subsystems managed by a single ART, and they reduce the cognitive load on stream-aligned ARTs, which are essentially consumers of that subsystem. For example, a navigation system's mapping and routing algorithms for autonomous driving could likely require the technical specializations of an entire complicated subsystem ART. The stream-aligned navigation ART would integrate this subsystem

and deliver it to the users as part of the larger system.

Platform ART

Similarly, it is common for a Solution Train to have Platform ARTs that provide services that the stream-aligned ARTs extend and build on. Continuing the autonomous vehicle example, a communication system that manages data transferred between the various subsystems would likely be built and maintained by a platform ART, providing clearly defined interfaces to its consumers.

Enabling ART

While not pictured in Figure 2 and less common than the other ART topologies, Enabling ARTs provide specialty tools, services, or expertise to other ARTs. Enabling ARTs can operate internally to help improve the Solution Train's ARTs delivery. Or they can operate externally, typically by enabling consumers of the Solution Train's large solution.

In all these examples, the ARTs comprise teams that also assume one of the four team topology types. For instance, the complicated subsystem ART developing the routing subsystem may have one or more stream-aligned teams building the [Features](#) that communicate navigation information to consumer components, with other complicated subsystem teams build the routing algorithms. In this manner, the application of the topologies is a fractal within the system.

Enabled by Critical Roles

In addition to the ARTs and Agile teams, the following roles aid the Solution Train's successful execution:

[Solution Management](#) defines and supports building desirable, feasible, viable, and sustainable large-scale business solutions that meet customer needs over the solution's significant lifespan. They represent the customer and business needs to the ARTs.

[Solution Architects](#) define and communicate a shared technical and architectural vision across the Solution Train to help ensure the solution under development is fit for its intended purpose. They work with the ART's System Architects to help guide their portion of the solution's design.

[Solution Train Engineer \(STE\)](#) is the coach for the Solution Train, facilitating and guiding the work of all ARTs and suppliers. The STE works with [Release Train Engineers \(RTEs\)](#) to facilitate ART execution and coordinate delivery.

[Suppliers](#) are internal or external organizations that develop and deliver components, subsystems, or services, which help Solution Trains deliver solutions to customers.

[Business Owners](#) are key stakeholders of the Solution Train, with final responsibility

for the business outcomes. They, along with Solution Train leaders, may also serve as business owners for the Solution Train's ARTs.

Customers are the buyers of the solution and ultimately determine value. When delivering in a supply chain, customers work closely with Solution Management and other key stakeholders to define and adjust the solution's vision, intent, and delivery roadmap.

Due to their size and cost, large solution development attracts a lot of attention, heightening stakeholder involvement and increasing opinions, governance, and oversight. The Solution Train leadership team – Solution Managers, Architects, and STE – must be aligned and represent a consistent force to realize the solution's vision. The STE should ensure this team has the time and space to form as a team and build cross-role transparency and trust.

The following roles also play an essential part in the Solution Train's success:

A **System Team** is typically formed for the Solution Train to address the integration issues across the ARTs. Individual ARTs may have their own Systems Teams. All System Teams share resources and align on common infrastructure and toolchains where possible.

Shared Services represent the specialty roles, people, and services required for the success of the Solution Train, but that cannot be dedicated full-time. Sometimes, they may be devoted to the Solution Train and shared across the ARTs.

Solution Train Responsibilities

ARTs within a Solution Train fulfill the responsibilities as described in the ART article. Since the ARTs and teams deliver most of the Solution Train's value, Solution Train leaders must support the ARTs in their value delivery. In doing so, Solution Trains also have their unique responsibilities, as shown in Figure 3.

Figure 3. Solution Train Responsibilities

Figure 3. Solution Train Responsibilities

Connecting with the Customer

Feedback is critical to value delivery. Inadequate feedback can result in high costs, rework, major delays, and unsatisfied customers. However, getting feedback presents unique challenges in large solution development. Significant organizational barriers often exist between the people building the solution and those who can provide the best input and feedback. To achieve overall alignment and enable decentralized decision-making, leaders help teams and suppliers building the system have direct conversations with key stakeholders that can provide the proper feedback to learn and adjust. The [Solution Train Flow](#) article offers meaningful guidance on the challenges and mechanisms for getting fast feedback when building these systems of scope and scale.

Planning and Roadmapping

No event is more powerful in SAFe than [PI Planning](#). Solution Trains must ensure that their ARTs enter PI Planning aligned on a shared vision, goals, and objectives

for the upcoming Planning Interval (PI). They perform [Pre-Planning](#) activities to prepare the ARTs and suppliers for PI Planning and foster alignment during the individual ART planning events.

Solution Train leaders use rolling-wave [Roadmaps](#) to align ARTs and suppliers and inform the features in their ART backlogs and PI roadmaps (Figure 4). Some ART features directly implement Solution Train [Capabilities](#) (discussed below). Other features come from a local context, driven by the milestones and releases in the solution roadmap.

Figure 4. A Solution Train roadmap showing multiple planning horizons

Figure 4. A Solution Train roadmap showing multiple planning horizons

Building Solution Capabilities

While some features come from local ART context, others implement the Solution Train's capabilities. Capabilities represent a higher level of solution behaviors that typically take multiple ARTs to implement, as shown in Figure 5. Like features, capabilities are also sized to fit within a PI.

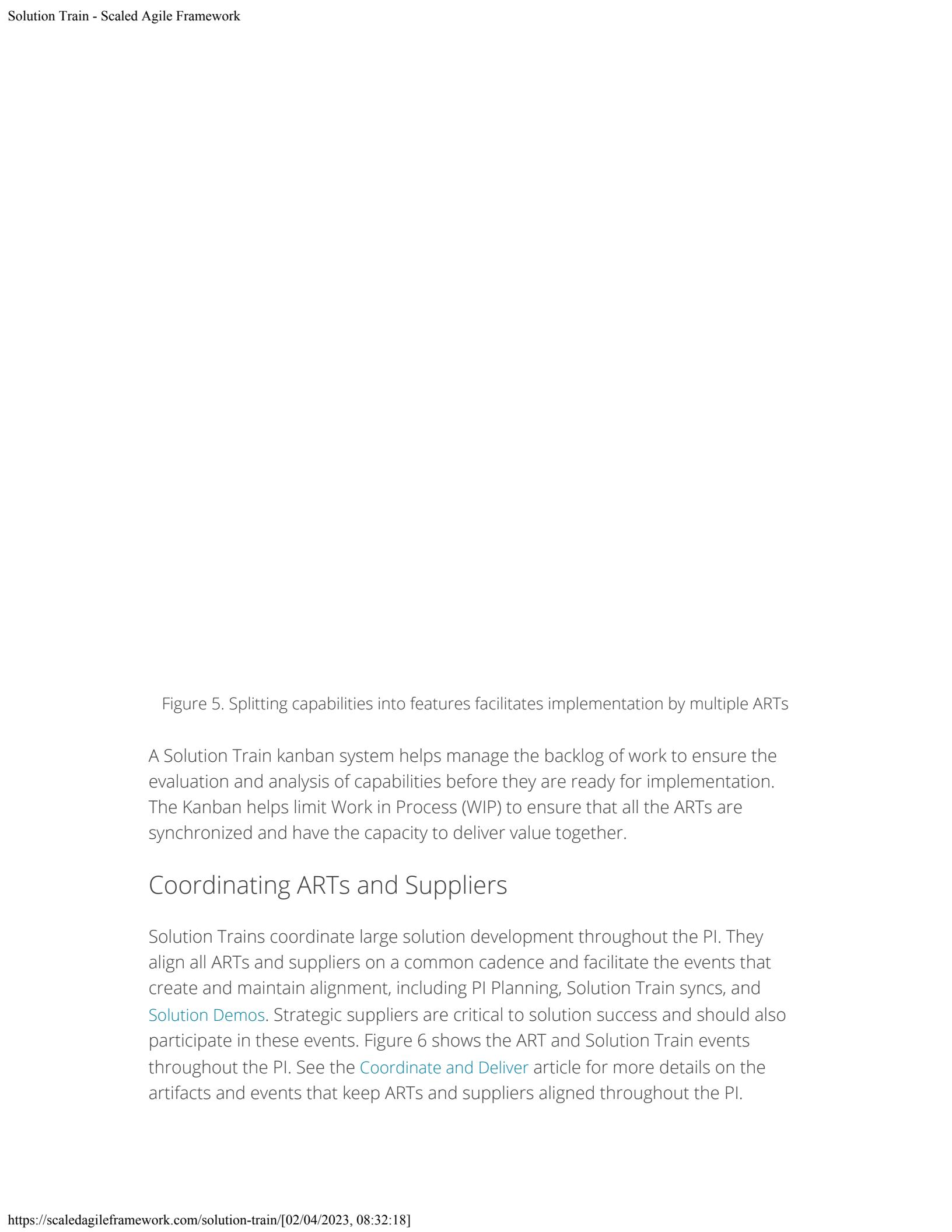


Figure 5. Splitting capabilities into features facilitates implementation by multiple ARTs

A Solution Train kanban system helps manage the backlog of work to ensure the evaluation and analysis of capabilities before they are ready for implementation. The Kanban helps limit Work in Process (WIP) to ensure that all the ARTs are synchronized and have the capacity to deliver value together.

Coordinating ARTs and Suppliers

Solution Trains coordinate large solution development throughout the PI. They align all ARTs and suppliers on a common cadence and facilitate the events that create and maintain alignment, including PI Planning, Solution Train syncs, and [Solution Demos](#). Strategic suppliers are critical to solution success and should also participate in these events. Figure 6 shows the ART and Solution Train events throughout the PI. See the [Coordinate and Deliver](#) article for more details on the artifacts and events that keep ARTs and suppliers aligned throughout the PI.

Figure 6. Solution Train coordination events

Figure 6. Solution Train coordination events

Releasing and Release Governance

Solution Trains apply cadence and synchronization to manage development. But, like ARTs, Solution Trains can deploy the solution, or an element of the solution, at any time the business and market dictates.

Ideally, teams and ARTs should be able to release at will for fast feedback, a flow of value delivery, and to learn in an operational environment. Automated testing, verification, and validation in the [Continuous Delivery Pipeline](#) (CDP) should address many release management quality concerns. However, releasing value in large solution development is often a significant governance concern as failures in these systems can have unacceptable social and economic costs. In addition, these systems are often subject to regulatory oversight and compliance, which can add additional constraints and delays to the release process.

Each Solution Train must establish—nor operate within the governance of—a release management function with the authority, knowledge, and capacity to foster and

approve releases. The release management function includes representatives from the Solution Train and ARTs. It may also include representatives from marketing, quality, Business Owners, IT Service Management, operations, deployment, and other areas. The release management team usually meets regularly to evaluate the solution's content, progress, and quality as it progresses to release. Even then, some releases will require even more extensive work and collaboration, such as communicating with customers or approval from external regulatory bodies.

Learn More

- [1] Skelton, Mathew, and Manuel Pais. *Team Topologies*. IT Revolution Press, 2019.
- [2] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

Last update: 23 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)
[Blog](#)

TRAINING

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL
[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

[Course Calendar](#)

Weekdays: 9am to 5pm

[About Certification](#)

Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



A shared vision is not an idea—it is rather, a force in people’s hearts—at its simplest level, a shared vision is the answer to the question “What do we want to create?”

—Peter Senge

[Adjust Cookie Settings](#)

Solution Management

Solution Management is the function responsible for defining desirable, viable, feasible, and sustainable large solutions that meet customer needs and for supporting development across the solution life cycle.

Details

The Solution Management role is key to successful enterprise solution delivery. Defining the solution [Vision](#), aligning the Solution Train and its stakeholders to the vision, and coordinating development progress across multiple ARTs and Suppliers is a challenging and critical task for every enterprise. But as is the case with [Product Management](#), the role of Solution Management for a Solution Train is typically fulfilled by multiple people, each guiding different aspects of the solution.

Key Collaborations

Creating a successful solution requires collaboration across the different roles and groups involved in development. Solution Management leads the interactions with the following participants (Figure 1):

Figure 1. Key interactions that involve Solution Management

Figure 1. Key interactions that involve Solution Management

The most critical interactions appear in the following areas:

Steer the Solution Train ▷ Three key roles guide the progress of solution development. Solution Management determines the direction of the solution.

[Solution Architects](#) provide the design that supports the business direction. [Solution Train Engineers \(STEs\)](#) facilitate the development and delivery. The interactions among these three roles occur at PI boundaries and during the PI execution.

Align on outcomes ▷ A live solution has to perform its intended function for the [Customer](#) while fulfilling the business goal for the organization. Solution Management works directly with the customer to ensure solution efficacy in solving the customer's need. Additionally, Solution Management remains aligned with [Lean Portfolio Management \(LPM\)](#) and other stakeholders to ensure that the solution achieves the enterprise's business objectives.

Advance the solution ▷ To define and elaborate solution [Capabilities and Features](#), Solution Management works side-by-side with Product Management and [Suppliers](#). This ongoing collaboration culminates at every PI boundary where Solution Management leads Product Management and suppliers as they prepare work

content for the new PI.

Responsibilities

Solution Management's responsibilities fall into five main areas, as shown in Figure 2.

Figure 2. Areas of responsibility of Solution Management

Figure 2. Areas of responsibility of Solution Management

This article further describes each of these areas.

Connecting with the Customer

Large solutions often serve multiple user personas and account for various use-case scenarios. Understanding the customer's need is essential to building a viable solution. In this regard, Solution Management does the following:

Determine customer needs and personas ☰ Solution Management determines what user personas the solution will benefit from and which tasks it will enable. This process frequently relies on continual research that feeds into the definition of new solution train backlog items. Often, [Design Thinking](#) facilitates exploring personas and uncovering their specific solution needs.

Understand solution context ☰ Solution Management determines which aspects of [Solution Context](#) are critical to solution development. To thoroughly understand the solution context, Solution Management works with architects and other technology subject-matter experts to navigate the complexities of the solution deployment and execution environment. To understand the operational context, Solution Management also interacts with the participants of the [Operational Value Streams](#) that the solution is serving. For solutions in active development, the solution context may often change as new capabilities are introduced, new user personas must be supported, or new infrastructure parameters are developed.

Measure solution usage ☰ Solution Management relies on facts to determine the course of action for the solution. Important data points come from solution telemetry that reveals user actions as a part of use-case scenarios realized in the solution. To generate this data, Solution Management proactively collaborates with Product Management of ARTs to define and instrument the necessary measures.

Defining the Large Solution

Defining the solution is a multilayered process that elaborates the vision with the detail needed to clarify what solution builders will create and deliver to the customer. As the Solution Train navigates the unknowns in the process, this detail is progressively elaborated.

Define and evolve the Solution Intent ☰ Solution Management leads the effort in defining and maintaining the [Solution Intent](#). This work involves multiple roles, including Product Management, architects, and other subject-matter experts from ARTs and suppliers.

Manage solution hypotheses, MVP, and leading indicators ☰ In SAFe, large solution development leverages accelerated learning. This involves building a Minimum Viable Product (MVP) for exploration and experimentation to realize a business hypothesis. Solution Management defines measurement criteria to support the hypothesis and—depending on the experiment’s outcome—exercises the decision to pivot or persevere. During solution development, multiple MVPs may validate different assumptions at different times.

Define Solution Epics and Capabilities ☰ A recurring task of Solution Management is to define the upcoming work for the solution train. The definition

process is realized via the [Solution Train Backlog](#) and involves multiple steps that support additional elaboration. However, not all the work content for a solution train is defined by Solution Management. ARTs can also leverage decentralized decision-making and specify work based on their direct knowledge of the customer context.

Elaborate solution definition with ARTs and suppliers ☰ Solution Intent and solution capabilities get further elaborated with ARTs and suppliers, accounting for more detail in the systems' behavior, architecture, and implementation constraints. This elaboration is usually performed leading up to the PI planning.

Planning and Managing the Solution Roadmap

Solution Management sets the direction for the solution. This helps the Solution Train and its stakeholders understand where solution development is headed and its progress over time. The path is continually adjusted to leverage new learnings and capitalize on new opportunities to deliver business value. The following activities are essential:

Align with portfolio stakeholders ☰ Solution Management ensures that the solution vision reflects the portfolio strategy and accommodates the portfolio initiatives that affect the Solution Train. Besides directly involving stakeholders in the activities that lead to the solution train's PI planning, Solution Management usually participates in sessions supporting the [Portfolio Backlog](#) and Kanban system.

Build and maintain solution roadmap ☰ To align ARTs, suppliers, and stakeholders to the desired solution progression over time, Solution Management creates and maintains the solution [Roadmap](#). The roadmap provides sufficient specificity in the near term to guide the development effort. However, specificity decreases for more distant future capabilities on the roadmap. This allows Solution Management to preserve options and leverage opportunities as they present themselves. At a minimum, the roadmap is updated at every PI boundary and presented to the Solution Train as a part of PI planning.

Manage Value Stream KPIs ☰ The ultimate purpose of any solution is to achieve desired outcomes for the customer and the business. Collaborating with portfolio stakeholders, Solution Management defines the [Value Stream KPIs](#) that are used to measure the business outcomes.

Managing and Prioritizing the Solution Train Backlog

The flow of customer value requires clear priorities: the Solution Train's ARTs and suppliers need a definitive idea of the next set

of capabilities they will be building. Knowing the priorities is crucial to understanding what needs to be built next. It also permits tradeoffs as capacity, technology, and other constraints reveal themselves.

Continually prioritize the Solution Train Backlog ☰ At every PI boundary, Solution Management makes critical decisions regarding work priorities for the Solution Train. Prioritization and elaboration work happens continually as new work items emerge or new facts are revealed.

Participate in Pre- and PI Planning ☰ Solution Management leads the preparation of work content for the Solution Train to perform a productive [Pre-Planning](#) and [PI-Planning](#). Solution Management presents the vision and roadmap to the train and participates in the planning, guiding as necessary on different aspects of the solution definition.

Define and manage capacity allocations ☰ Solution Management determines capacity allocations for the Solution Train, typically including capacity for developing new business capabilities, architectural enablers, or performing maintenance work. Additionally, they help determine how much capacity ARTs should dedicate to Solution Train backlog items versus work that emerges locally.

Working with ARTs and Suppliers to Deliver Value

Large solution development relies on effectively coordinating solution development across many ARTs and suppliers. A big part of this coordination is defining and refining what to build. Inherently, this process involves adjustment and adaptation based on an ongoing, objective evaluation of the working solution.

Participate in release governance ☰ Solution Management plays a vital role in approving releases and influencing their timing and scope. It's actively involved in planning releases and making adjustments as needed. Additionally, they participate in defining the release-level definition of done.

Regularly review integrated solution increments ☰ With Agile, development progress is based on working systems. Solution Management regularly reviews integrated increments of value and makes necessary adjustments to development plans.

Coordinate value delivery across ARTs and suppliers ☰ As multiple ARTs and suppliers are involved, a significant part of the Solution Management effort is ensuring that each participant in the development process is moving toward a common goal. Solution Management leads the Product Management sync multiple times per PI. A Solution Train Planning Board (see [Coordinate and Deliver](#)) is used to track the progress of business capabilities across suppliers and ARTs.

Participate in I&A with ARTs and suppliers ☰ Solution Management participates in the Solution Train [Inspect and Adapt](#) (I&A) to review and act on opportunities for improvement. This allows Solution Management to hear firsthand the challenges that the Solution Train experiences along with the ideas on how to address them.

Last Updated: 13 February 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit [Permissions FAQs](#) and contact us for permissions.

FRAMEWORK	PARTNER	CONTENT & TRADEMARKS	SCALED AGILE, INC
Download SAFe Posters & Graphics	Becoming a Partner Partner Directory	FAQs on how to use SAFe content and trademarks Permissions Form Usage and Permissions	CONTACT US 5400 Airport Blvd., Suite 300 Boulder, CO 80301 USA
Watch and download SAFe videos and presentations	GET SOCIAL		
Blog	Twitter Linkedin YouTube		
TRAINING			BUSINESS HOURS
Course Calendar About Certification Become a Trainer			Weekdays: 9am to 5pm Weekends: CLOSED
			Privacy Policy Cookie Policy

English

Scaled Agile Framework



*A system is never
the sum of its parts;
it's the product of
their interaction.*

—Russel Ackoff

Solution Train Engineer

The Solution Train Engineer (STE) is a servant leader and coach who facilitates Solution Train events and processes, coordinates the work of ARTs and Suppliers, and supports ARTs in delivering value.

Adjust Cookies Settings

To succeed in the digital age, many organizations must efficiently build and evolve **large solutions** that fulfill their customers' most critical business needs. Building such solutions can involve hundreds—sometimes thousands—of people on **Solution Trains** organized around complex value streams. These value streams require a dedicated role to lead associated **Agile Release Trains** (ARTs) and **Suppliers** through the efficient execution of their shared solution strategy. That is the role of the Solution Train Engineer (STE).

Details

Leading Solution Trains through the delivery of large solutions requires significant coordination of the end-to-end value stream. This is challenging because a Solution Train involves multiple ARTs and Suppliers that must operate in unison across distinct product lines and organizational boundaries. The STE is one of a trio of roles (Figure 1) that aligns people and teams to a shared solution strategy.

Figure 1. The Solution Train Ôtriad

The roles reinforce one another. **Solution Management** and **Solution Architects** define the solution, while the STE coordinates efficient development and delivery. Together, they steer the Solution Train through the multiple plan-do-check-adjust cycles (see **PI** and **Iterations**) required to deliver large solutions in the shortest sustainable lead time.

The STE also works in close collaboration with the **Agile Release Train** (ARTs) **Release Train Engineers** (RTEs), each of whom is responsible for coordinating the delivery of a critical aspect of the solution. Together, the STE and RTEs comprise a coordination and facilitation network that guides all teams on the Solution Train through the effective execution of the solution strategy.

Responsibilities

STE responsibilities fall into the five areas shown in Figure 2. Each is further elaborated in the sections below.

Figure 2. STE areas of responsibility

Figure 2. STE areas of responsibility

Facilitating Solution Train PI Planning

ARTs on a Solution Train plan their activities through a series of common, cadence-based PI Planning events. Aggregating those events results in a PI Plan for the large solution. The ART planning events may be conducted as a single "Solution Train PI Planning" event, or ARTs may plan in a more distributed fashion. In either case, Solution Train PI Planning is a critical synchronization point across all the ARTs and suppliers. It requires careful preparation and execution. It leverages many input sources, generates many outputs, and includes a variety of business and technical stakeholders. The STE ensures that PI Planning is steered by high-quality information and that strategic alignment among stakeholders is achieved.

Several weeks before reaching the PI planning boundary, the STE begins to prepare for PI Planning by performing the following activities, which contribute to the Solution Train's broader [Pre-Plan](#) strategy:

Update the solution vision and roadmap ▷ The solution [Vision](#) and [Roadmap](#) provide essential context for PI Planning. STEs collaborate with Solution Management, Solution Architects, and others as necessary to ensure they represent an accurate summary of the overall solution strategy. This strategy must also be reflected in the [Solution Train Backlog](#) as prioritized [Capabilities](#) and guided by well-balanced capacity allocation.

Gather feedback and metrics ▷ To fuel relentless improvement, the STE collects evidence of the Solution Train's recent performance toward target outcomes, competency, and flow. The STE works with Solution Management and Solution Architects to obtain quantitative and qualitative insights from customers and systems.

Assist with PI Planning logistics ▷ As a matter of final preparation, the STE works with RTEs, stakeholders, and other ART representatives as necessary to schedule PI planning events, invite key participants, and communicate the agenda. If needed, the STE facilitates a Pre-Plan workshop immediately before PI planning to achieve alignment among Solution Train leaders and stakeholders.

During PI planning, the STE focuses on the following to ensure productive outcomes are achieved:

Synchronize with ARTs ▷ STEs maintain reliable communication with RTEs to sense and address impediments across ARTs quickly. This often entails facilitating several RTE syncs utilizing a solution planning board as the primary information radiator.

Address uncertainty ▷ Unforeseen dependencies and information gaps often arise during PI planning. To dampen the effects of these surprises, the STE connects teams with the information sources needed to resolve them quickly.

Define Solution Train PI objectives ▷ Solution Train PI planning aims for all participating ARTs and suppliers to commit confidently to a joint plan. The STE creates Solution Train [PI Objectives](#)—often as an aggregation of ART PI objectives—to represent this plan.

Coordinating Large Solution Delivery

Cadence-based delivery is the core of Solution Train operations. Successful delivery of large solutions requires a series of learning cycles throughout PI execution. The STE is instrumental in coordinating

these learning cycles, which maintains alignment among all Solution Train participants and enables the rapid, continuous delivery of value to customers.

During PI execution, the STE prioritizes the following activities, which are described in detail in the [Coordinate and Deliver](#) article:

Facilitate synchronization events ⚡ Regular syncs keep the Solution Train on track. STEs are key members and facilitators of RTE Syncs and may organize Solution Train Syncs when the needs of RTEs, Product Management, and Solution Architecture overlap.

Ensure frequent solution integration ⚡ Frequently integrating the solution's parts reduces risk and supports steady development progress. The STE ensures that the Solution Train values frequent integration and helps ARTs and [System Teams](#) succeed.

Facilitate solution demo ⚡ Solution Trains do not typically produce an end-to-end [Solution Demo](#) at every iteration. However, the STE frequently helps the Solution Train showcase the integrated—or partially integrated—solution throughout the PI.

Coordinate release activities ⚡ STEs work with RTEs, Solution Management, and [Product Management](#) to oversee release activities for the Solution Train. Significant events on the solution roadmap and the solution planning board guide release timing.

Coaching Solution Train Stakeholders

The STE serves as a model of [Lean-Agile Leadership](#) to the Solution Train, reinforcing the Lean-Agile mindset, values, and principles. The STE coaches leaders directly but collaborates more informally with other roles to increase the organization's [Enterprise Solution Delivery](#) (ESD) competency.

The STE focuses on the following activities to coach the Solution Train to sustained proficiency:

Lead by example ⚡ STEs lead change and gain earned authority by modeling Lean-Agile behaviors. STEs operate with a growth mindset and deep respect for

transparency, people, alignment, and improvement. Additionally, they are guided by SAFe principles and the three domains of ESD.

Coach fellow Solution Train leaders Ø STEs help RTEs, [Epic Owners](#), [Business Owners](#), enterprise executives, supplier stakeholders, and other influential Solution Train members lead by example. They do this by holding leaders responsible for their actions, assisting with building Agile leadership teams, and coaching leaders away from traditional behaviors, such as siloed decision-making, adherence to fixed plans, cost-based accounting, and phase-gated delivery.

Foster a continuous delivery culture Ø STEs collaborate with Solution Management, Solution Architects, RTEs, Product Management, System Architects, and Business Owners to establish Lean Systems Engineering practices (see [ESD](#)), build [Continuous Delivery Pipelines](#), and architect for flexibility across the Solution Train landscape.

Build influential relationships Ø Coordinating Agile delivery across a Solution Train is complex and can be impeded by politics, communication challenges, and philosophical debates. To maintain widespread alignment and shared purpose, STEs build authentic relationships with key influencers within and outside the Solution Train and serve them as trusted advisers.

Optimizing Flow

As described in [Principle #6](#) Ø Make value flow without [interruptions](#), SAFe is a flow-based system that understands and addresses eight immutable flow properties. Each of these properties—batches, handoffs, and policies, to name a few— influences the efficient movement of work through the system. Because Solution Trains tend to have complex value streams that span ARTs, suppliers, and product lines, flow can quickly break down if not carefully managed. In an ongoing effort to maximize [Solution Train Flow](#), STEs perform the following activities:

Facilitate value stream mapping Ø The STE collaborates with RTEs, stakeholders, suppliers, and subject matter experts to visualize each value stream that pertains to the Solution Train. This clarifies the actual, end-to-end flow of activities across all teams, functions, and organizational boundaries and aligns leaders across the Solution Train on the health of the entire delivery system.

Measure Solution Train flow Ø As described in the [Measure and Grow](#) article, SAFe provides six flow metrics that gauge how effectively an organization delivers value. STEs apply them to large solution delivery, establishing a holistic view of flow time, distribution, velocity, load, efficiency, and predictability across the Solution Train. Baseline measurements are established during value stream mapping, then revisited iteratively to track improvements.

Apply flow accelerators ☰ The combination of value stream mapping and flow measurements often reveals significant delays in the delivery process. These delays often stem from mistimed inter-ART handoffs, supply chain bottlenecks, large development batches, and legacy governance policies. The STE guides Solution Train stakeholders in applying SAFe's flow accelerators to streamline efficiency across the entire delivery system.

Improving Relentlessly

Developing and delivering large solutions iteratively and reliably is crucial for Solution Trains. Developing this capability requires an ongoing commitment to learning and improving. Like any part of a SAFe organization, a Solution Train evaluates its effectiveness across the outcomes, flow, and competency measurement domains (Figure 3).

Figure 3. The three measurement domains of SAFe

Figure 3. The three measurement domains of SAFe

Appreciating that measurements are derived from a variety of sources, including customers, the STEs:

Identify meaningful metrics Ⓛ Not all metrics are created equal, nor do they have the same relevance simultaneously. The STE is instrumental in identifying the metrics that apply best to the Solution Train at any given time and working with Solution Train stakeholders and peers to establish meaningful improvement targets. (Simply aggregating ART-level performance indicators does not paint a complete picture of Solution Train performance.)

Measure often Ⓛ The collection and evaluation of measurements should coincide with a specific plan-do-check-adjust cycle to inform a crucial decision at the right time. The STE assists with gathering measurements at the appropriate intervals—per day, per iteration, per PI, or otherwise—to provide the most valuable feedback at the most valuable cadence.

Facilitate solution level I&A Ⓛ Solution Trains [Inspect & Adapt \(I&A\)](#) to address impediments and problems that span ARTs and suppliers. STEs are crucial facilitators of this event, ensuring that discussions lead to global improvements rather than ones that locally optimize individual ARTs or teams.

In addition, the [Enterprise Solution Delivery assessment](#) reveals the current strengths and challenges of the Solution Train across several dimensions of competency. It is a logical starting point for identifying meaningful improvement activities.

Last update: 14 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use](#)

SCALED AGILE, INC

[CONTACT US](#)

Watch and download
SAFe videos and
presentations

Blog

TRAINING

Course Calendar

About Certification

Become a Trainer

GET SOCIAL

Twitter

Linkedin

YouTube

SAFe content and
trademarks

Permissions Form

Usage and Permissions

5400 Airport Blvd., Suite

300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

Privacy Policy

Cookie Policy

Your California

Consumer Rights

English

Scaled Agile Framework



Agility is principally about mindset, not practices.

— Jim
Highsmith, Agile
Project Management:
Creating Innovative
Products

Lean-Agile Mindset

The Lean-Agile Mindset is the combination of beliefs, assumptions, attitudes, and actions of SAFe leaders and practitioners who embrace the concepts of Lean

Thinking and the Agile Manifesto.

It's the personal, intellectual, and leadership foundation for adopting and applying SAFe principles and practices.

The Lean-Agile mindset forms the cornerstone of a new way of working and an enhanced company culture that enables Business Agility. It provides leaders and change agents with the tools needed to drive a successful SAFe transformation, helping individuals and enterprises achieve their goals.

Details

Mindset Awareness and Openness to Change

What exactly is a "mindset"? A mindset is a mental lens through which we view the world around us. It is how the human brain simplifies, categorizes, and interprets the vast amount of information it receives daily. We form our mindsets through a lifetime of structured learning (classes, reading) and unstructured lessons (life events, work experience). They reside in the subconscious mind and manifest themselves as deeply held beliefs, attitudes, assumptions, and influences. Consequently, individuals are often unaware of how their mindsets influence how they carry out their responsibilities and interact with others. While many mindsets are positive and serve us well, others may need to change over time [1].

So how can mindsets be changed? It begins with an awareness of one's current mindsets and how they were formed. It's also vital to cultivate the belief that mindsets can be developed and improved (a "growth" mindset, as illustrated in Figure 1).

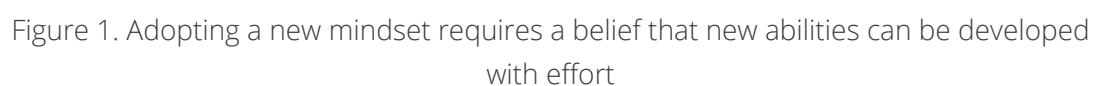


Figure 1. Adopting a new mindset requires a belief that new abilities can be developed with effort

Changing mindsets is a vital topic in transitioning to SAFe because, too often, leaders and practitioners in organizations go through the motions of mimicking SAFe practices and using SAFe terms without internalizing and embracing the underlying values and principles that truly represent a new way of working. This ÔSAFe in name onlyÕ approach may produce some small successes in the short term. However, in the long term, such a shallow adoption of the Lean-Agile mindset will inevitably fail to produce the real, long-lasting business results leaders hoped for when they decided to Ôgo SAFe.Õ

To fully embrace SAFe requires a ***growth*** mindset open to learning the core values and principles of two primary underlying bodies of knowledge: Lean Thinking and Agile. Each has a rich and deep history of published guidance and case studies. Their respective values and principles need to be understood and practiced so that the ideals of both Lean and Agile permeate the organizationÕs language, practices, and decision-making. Ultimately, it simply becomes Ôour way of workingÕ and is deeply ingrained in the culture of the enterprise.

The following two sections describe the key elements of Lean Thinking and Agile (summarized in Figure 2) that form the basis of the Lean-Agile mindset.

Figure 2. Lean Thinking and Agile are the core building blocks of SAFe

Figure 2. Lean Thinking and Agile are the core building blocks of SAFe

Lean Thinking



Adopting Lean product development can double labor productivity through the entire system, cut time-to-market for new products in half, and enable a wider variety of products within product families to be offered at very modest additional cost.

From *Lean Thinking* by Womack & Jones

Initially derived from Lean manufacturing, the principles and practices of Lean thinking as applied to software, product, and systems development are now deep and extensive [2]. For example, Ward [3], Reinertsen [4], Poppendieck [5], Kersten [6], Leffingwell [7], and others have described aspects of Lean thinking, placing many of the core principles and practices within a product development context. Applying Lean Thinking to product development, thereby shifting from the traditional batch-and-queue production system to continuous flow with an effective pull by the customer, can lead to dramatic improvements.

Lean thinking can be summarized as shown in Figure 3:

Figure 3. The core tenets of Lean Thinking

Figure 3. The core principles of Lean Thinking

As Figure 3 illustrates, the goal of Lean Thinking is to deliver the maximum value (a solution) to the customer in the shortest sustainable lead time from the trigger (the identification of the need or opportunity) to the point at which the customer receives the value. How value is created also matters. High quality, respect for people and society, high morale, safety, and customer delight are also essential goals and benefits of Lean Thinking. Achieving these goals requires applying the five basic principles of Lean, illustrated in Figure 3 and described in the following sections.

Precisely specify value by specific product

Every enterprise is built to deliver value. Value can only be defined by the ultimate customer. And it's only meaningful when expressed in terms of a specific product (a good or a service, and often both at once) that meets the customer's needs at a specific price at a specific time [8]. Therefore, the first principle of Lean Thinking underscores the importance of understanding customers' needs and quantifying the value inherent in the solutions delivered to them. The solution itself holds the value—not the project, initiative, or process that produces it—and the customer ultimately determines that value.

Identify the Value Stream for each product

Once "value" is defined for each product and type of customer, the following principle in Lean Thinking is to articulate how the enterprise creates that value, from identifying a need or opportunity to delivering the solution. This flow of work is the value stream and contains all the people, processes, tools, and information necessary to deliver value. Delays anywhere in this system result in delayed delivery of value to customers.

Make value flow without interruptions

The third principle in Lean Thinking is establishing a continuous, uninterrupted flow of work that supports incremental value delivery based on constant feedback and adjustment. Enabled by [Built-In Quality](#) practices, relentless improvement, and evidence-based governance, continuous flow enables faster, sustainable value delivery.

Achieving a continuous flow of value requires applying and understanding the eight fundamental properties of flow: visualizing and limiting work-in-process (WIP), addressing bottlenecks, minimizing handoffs and dependencies, getting fast feedback, working in small batches, managing queue lengths, optimizing time "in the zone," and remediating legacy policies and practices. These flow properties are described in greater detail in the [SAFe Principle 6](#) article and in the [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#) articles. The SAFe [Core Values](#) and [SAFe Principles](#) help teams achieve a continuous flow of value at scale in large, complex enterprises.

Let the customer pull value from the producer

The next Lean principle guides organizations to configure value streams to deliver solutions that customers *pull* into the market based on their actual needs rather than solutions that teams *push* into the market based on what they "think" customers need. This is key to calibrating the capacity of the value stream. Too much capacity compared to market pull results in waste, which is antithetical to

Lean thinking. Conversely, too little capacity creates bottlenecks and delays, defeating the intent to provide the customer with a continuous flow of value.

Pursue perfection

The final principle of Lean Thinking is expressed as “pursue perfection.” It reflects that no matter how closely the first five principles are followed, creating a fast and effective flow of value is not a one-time activity. Market dynamics, customer needs, and available technologies are just some of the many factors that can require value streams to be refined and, in some cases, rebuilt entirely.

Agile



Agile is an attitude, not a technique with boundaries. An attitude has no boundaries, so we wouldn't ask “can I use Agile here,” but rather “how would I act in the Agile way here,” or “how Agile can we be, here?”

— Alistair Cockburn

In the 1990s, some lighter-weight and more iterative development methods emerged in response to the many challenges of waterfall processes. In 2001, many leaders of these frameworks came together to express their shared values and beliefs in the Manifesto for Agile Software Development. This turning point clarified the new approach and started to bring the benefits of these innovative methods to the whole development industry. [8] Since the Manifesto was first published, Agile has been adopted by domains outside of software development, including hardware systems, infrastructure, operations, and support. More recently, business teams outside of technology have also embraced Agile principles for planning and executing their work.

The Values of Agile

Agile is built on the value statement shown in Figure 4. For the remainder of the description of Agile Values and Principles, as initially outlined in the Agile Manifesto, readers can expand each use of the term “software” to include the working output of any Agile team, regardless of the domain.

Figure 4. Agile Values

Figure 4. Agile Values

Individuals and Interactions over Processes and Tools

Deming notes, “If you can’t describe what you are doing as a process, then you don’t know what you are doing. [10]” So, Agile processes in frameworks like Scrum, Kanban, and SAFe matter. However, a process is only a means to an end. When we’re captive to a process that isn’t working, it creates waste and delays. So, favor individuals and interactions, then modify processes accordingly. Tools are valuable but should supplement, rather than replace, face-to-face communication.

Working “Software” over Comprehensive Documentation

Documentation is important and has value. But creating documents to comply with potentially outdated corporate governance models has limited value. As part of a change program, governance, often captured by documentation standards, needs to be updated to reflect the Lean-Agile way of working. Rather than create detailed documentation too early—especially the wrong kind—it’s more valuable to show customers working software, systems, and so on to get their feedback. Therefore, favor measuring progress by evaluating tangible work products. And document only what’s truly needed.

Customer Collaboration over Contract Negotiation

Customers are the ultimate deciders of value, so their close collaboration is

essential in pursuing business agility. Contracts are often necessary to convey each party's rights, responsibilities, and economic concerns—but recognize that contracts can over-regulate what to do and how to do it. They don't replace regular communication, collaboration, and trust, no matter how well they're written. Instead, contracts should be win-win propositions. Win-lose contracts usually result in poor economic outcomes and distrust, creating contentious short-term relationships instead of long-term business partnerships. Instead, favor customer collaboration over contract negotiation.

Responding to Change over Following a Plan

Change is a reality in the digital age and essential to achieving agility. The strength of Lean-Agile is in how it embraces change. As the system evolves, so does the understanding of the problem and the solution domain. Business stakeholder knowledge also improves over time, and customer needs evolve as well. Indeed, those changes add value to our system.

Of course, planning is an essential part of Agile. In fact, Agile teams and teams-of-teams plan more often and more continuously than their waterfall counterparts. However, plans must adapt as new learning occurs, new information becomes visible, and the situation changes. Worse, evaluating success by measuring conformance to a plan drives the wrong behaviors (for example, following a plan in the face of evidence that the plan is not working).

Agile Principles

Agile has 12 self-explanatory principles that support its values, as shown in Figure 5. These principles take Agile Values a step further and specifically describe what it means to be Agile.



Figure 5. Principles of the Agile Manifesto

The combination of these values and principles creates the essence of Agile. There is overwhelming evidence from success stories in all industries across every geography demonstrating the extraordinary business and personal benefits of this new way of thinking and working. We are grateful for it.

Applying Lean Thinking and Agile in SAFe

Collectively, the values and principles of Lean Thinking and Agile form the DNA of everything contained within SAFe. All the roles, practices, events, and artifacts in SAFe are designed to provide practical guidance for adopting the combination of these two bodies of knowledge as the new way of working throughout the enterprise.

Thousands of implementations of SAFe over the last decade have shown that Lean Thinking and Agile principles and practices have unique implications when applied at scale. For example, providing an uninterrupted flow of value within the context of a single Agile team will look different than when this same principle is applied to an entire portfolio. Yet the principle is equally important in both cases. The

implications of Lean and Agile at scale have been captured in the SAFe Core Values and SAFe Principles articles.

Learn More

- [1] Dweck, Carol S. *Mindset: The New Psychology of Success*. Random House Publishing, 2007.
- [2] Womack, James P., Daniel T. Jones, and Daniel Roos. *The Machine That Changed the World: The Story of Lean Production—Toyota’s Secret Weapon in the Global Car Wars That Is Revolutionizing World Industry*. Free Press, 2007.
- [3] Ward, Allen, and Durward Sobeck. *Lean Product and Process Development*. Lean Enterprise Institute, 2014.
- [4] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas, 2009.
- [5] Poppendieck, Mary, and Tom Poppendieck. *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley, 2006.
- [6] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. IT Revolution Press, 2018.
- [7] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.
- [8] Womack, James, and Daniel Jones. *Lean Thinking: Banish Waste and Create Wealth in your Corporation*. Free Press, 2003.
- [9] Manifesto for Agile Software Development. <http://AgileManifesto.org/>
- [10] Deming, W. Edwards. *Out of the Crisis*. MIT Center for Advanced Educational Services, 1982.

Last update: 6 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The restriction to these four team types acts as a powerful template for effective organization design.

—Matthew Skelton &
Manuel Pais

Organizing Agile Teams and ARTs: Team Topologies at Scale

[Adjust Cookie Settings](#)

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Overview

SAFe Principle #10 Ð [*Organize around value*](#), guides enterprises to organize people and teams around one goal: continuously delivering value to the customer. But to do so, they must consider how best to design their [*Agile Teams*](#) and [*Agile Release Trains \(ARTs\)*](#). Traditionally, this has been accomplished in various ways: organizing around features, components, sources of funding, even geography, etc. Each approach aims to bring people and cross-functional skills together to enhance flow, the throughput of value, and even the joy of work.

Organizing by [*Feature*](#) and component has been the standard approach for teams and trains within SAFe and Agile more generally.

In their book *Team Topologies*, Matthew Skelton and Manuel Pais have advanced the practices for team design. Specifically, they provide insights on organizing people around four fundamental team ÐtopologiesÑ: *stream-aligned*, *complicated subsystem*, *platform*, and *enabling* teams. [1] Each team topology has specific behaviors and responsibilities. By restricting team design to the four fundamental *team topologies*, the organization can use proven patterns to promote flow and deliver value faster.

This article describes these team topologies and applies them to SAFe. Doing so provides new and enhanced scaling patterns for organizations developing even the largest and most complex software and cyber-physical systems.

Details

For context, any [*Solution*](#) can be thought of from two perspectives:

1. The features define the value perspective it delivers to customers and end users.
2. The *technical* perspective is how the architectural components of the system interact to implement that functionality.

Organizing teams around features (feature teams) and components (component teams) has been the dominant pattern in Agile. [2] This focuses on each [*Agile Team*](#), orienting them to work and limiting their cognitive load to a single concern. In other words, they donÐt have to understand everything about the entire system, enabling them to focus on the part of the system they are responsible for.

However, this approach is not without challenges. The defining characteristics of a feature team are often unclear and do not always imply end-to-end value delivery. Additionally, the motivations for creating component teams are varied. Optimizations around specific technical expertise and software reuse typically drive those decisions. But this often results in too many teams aligned to specializations and technology, which increases dependencies and inhibits flow.

In their book, Skelton and Pais outline an alternative approach. They describe four types of teams that enhance and simplify the task of organizing around value (Figure 1):

Figure 1. The four fundamental team topologies

Figure 1. The four fundamental team topologies

Stream-aligned team Ø organized around the flow of work and can deliver value directly to the customer or end user.

1. **The complicated subsystem team** is organized around specific subsystems requiring deep specialist skills and expertise.
2. **Platform team** Ø organized around developing and supporting platforms that provide services to other teams.
3. **Enabling team** Ø organized to assist other groups with specialized capabilities and help them become proficient in new technologies.

No matter how they are organized, Agile Teams are the fundamental building blocks of SAFe, as almost everyone on the ART is part of a team. Each is a cross-functional group of ten or fewer individuals who define, build, test, and deploy an increment of value in a short timebox. The team includes a [Product Owner](#), who leads the definition and prioritization of team backlog, and a [Scrum Master/Team Coach](#), who acts as a servant leader. Together with this defined team structure, these topologies provide a clearer and better model for organizing Agile Teams in SAFe. Each team type and its responsibilities and behaviors are described below. They are designed to improve deployment frequency, team culture, and collaboration across teams and departments.

Stream-Aligned Teams

The term *stream-aligned* emphasizes the importance of organizing teams to deliver value continuously within the development value stream that builds, runs, and supports the product or solution. Skelton and Pais define a stream-aligned team as follows:

A stream-aligned team is aligned to a single, valuable stream of work, empowered to build and deliver customer or user value as quickly, safely, and independently as possible without requiring handoffs to other teams to perform parts of the work. [1]

One of the most significant benefits of organizing teams this way is [Customer Centricity](#). Each team has a direct relationship with the customers they serve. Stream-aligned teams apply [Design Thinking](#) practices to understand better the personas representing the customer segments they serve—building and supporting their desired features. It stands to reason that most teams in a Lean-Agile enterprise should be stream-aligned.

Rarely can a single stream-aligned team build an entire solution. More commonly, stream-aligned teams support a portion of the development value stream aligned to one of the following aspects:

- A specific solution or solution subset
- A set of features
- A specific customer persona
- A customer journey or particular steps in the journey
- A specific business domain
- Compliance and regulation requirements
- New product innovation

The determining factor is whether the stream-aligned team is responsible for

building and delivering customer value with minimal dependencies on other teams. This requires that stream-aligned teams be cross-functional and include all the skills necessary to develop and support whatever features and components they need. This also ensures that stream-aligned teams are long-lived, developing knowledge and creating efficiencies over extended periods.

Skelton and Pais define a set of expected behaviors for each team type. Within the context of SAFe, we interpret these responsibilities for stream-aligned teams as follows:

Know your customer Ⓛ build and maintain direct customer relationships and develop a deep understanding of the market segments served.

Develop a steady flow of new features describing a customer's need and the associated benefits. New features should make up most of the work stream-aligned teams deliver.

Apply Design Thinking Ⓛ understand the problem space, explore solution options, validate with customers, and incorporate feedback.

Apply continuous improvement practices Ⓛ reserve capacity for improving the processes and tools needed to do the work.

Build in quality Ⓛ ensure all work meets appropriate quality standards throughout development.

Collaborate Ⓛ identify and manage collaborations with other teams on the ART and shared services.

Respond to customer needs Ⓛ react to new feature requests and incidents and adjust the course of action.

Support the solution in production Ⓛ stream-aligned teams take responsibility for supporting their solution elements. In other words, "they build it, they run it."

Complicated Subsystem Teams

While it is reasonable to have primarily stream-aligned teams, it's unlikely that this will be the only team type required. As solutions become more extensive and complex—often including a mix of hardware and software components—they will likely include subsystems. Building and operating some of these subsystems requires specialist knowledge and expertise. Skelton and Pais acknowledge this by defining the purpose of a *complicated subsystem team*:

A complicated-subsystem team is responsible for building and maintaining a part of the system that depends heavily on specialist knowledge. Most team members must be specialists in that area of expertise to understand and make changes to the subsystem. [1]

Requiring stream-aligned teams to gain and maintain the requisite skills to the necessary proficiency levels across all potential subsystems would create too much

cognitive load. The teams can become overwhelmed by the complexity, unable to focus on a domain they can truly master. Instead, complicated subsystem teams pick up a significant portion of that load, taking responsibility for building and maintaining those parts of the system that require deep and ongoing technical expertise.

A complicated subsystem team could build things such as:

- Highly specialized system components are often used across multiple systems
- Safety-critical systems elements, which have a high cost of failure
- Specialty algorithms or business rules that are critical for fitness of use in the domain
- A part of a cyber-physical system (for example, an engine control module in an autonomous vehicle)

While all solutions can be decomposed into subsystems, not all require complicated subsystem teams. The level of expertise, complexity, and risk should be the *only* deciding factors for creating complicated subsystem teams.

This contrasts with traditional component teams, which may be justified for other good reasons, such as reuse or architectural integrity. As a rough guide, a single ART should contain no more than 1-3 complicated subsystem teams.

The responsibilities and behaviors of complicated subsystem teams include:

- Build, maintain, and support the complicated subsystem** Ⓛ recognize and commit to the critical technical elements they build.
- Maintain their level of expertise** Ⓛ continue to advance the knowledge and skills required to work within that subsystem domain.
- Collaborate with stream-aligned teams** Ⓛ ensure the subsystems are developed to meet customer requirements.
- Plan and prioritize effectively** Ⓛ align the subsystem roadmap with the needs of the stream-aligned teams.
- Develop appropriate interfaces** Ⓛ hide the complicated nature of the subsystems behind well-documented, easy-to-use interfaces.
- Takes responsibility for built-in quality** Ⓛ ensures the subsystem's quality, performance, and architectural robustness.

Platform Teams

A technology or computing *platform* is a set of services stream-aligned teams can access, typically via self-service APIs. Much like the complicated subsystem teams, platform teams (and the platforms they maintain) are created to reduce a stream-aligned team's cognitive load. Moreover, they should be allocated in a way that increases the autonomy of the stream-aligned teams. Platform teams are defined

as follows:

Platform team[s] provide the underlying internal services required by stream-aligned teams to deliver higher-level services or functionalities, thus reducing their cognitive load. [1]

This focus on platform teams as service providers heavily influences the way they work. Platforms are treated as products developed for their customers, the stream-aligned teams that utilize them. Customer centricity and design thinking apply in this context also. Additionally, their services should be well documented, easy to use, fit for purpose, narrow in scope, and offer reuse opportunities.

Responsibilities and behaviors of platform teams include:

Collaborate with stream-aligned teams □ ensure the platforms are developed per customer requirements.

Build the platform incrementally □ build and deploy in increments and secure frequent customer feedback and validation.

Focus on usability □ provide platforms that are easy to use via self-service capabilities and supporting documentation.

Commit to supporting and maintenance □ ensure the platform's sustainability and uptime, and commit to appropriate service-level agreements.

Lead by example □ keep the platforms thin and develop them on top of other platforms where applicable.

Enabling Teams

The tools and techniques for solution development are constantly changing, providing organizations with regular opportunities to integrate new practices and technologies. Although this brings many benefits, it also represents challenges to developing the necessary skills and expertise across all the teams. Enabling teams is an essential construct. They can provide support and guidance to other teams, assisting them in gaining these new skills and getting up to speed with these emerging technologies. Enabling teams are defined as follows:

Enabling teams helps stream-aligned teams acquire missing capabilities, usually around a specific technical or product management area. [1]

One example of an enabling team in SAFe is the [System Team](#), which assists ART teams with (among other things) building and supporting the [Continuous Delivery Pipeline](#). Some more specialized examples of enabling teams might provide expertise and support in the following areas:

DevOps implementation

- Automated testing
- Continuous integration and build tooling
- Engineering quality practices
- Security
- Environments and configuration

Enabling teams may also be focused on helping stream-aligned teams the first few times they need to integrate with a specific subsystem or platform. However, enabling teams are not there to fix quality issues caused by stream-aligned teams. Instead, they work with them for short periods, typically for a PI or so, to increase their skills and embed the required capabilities.

Depending on their charter, enabling teams may be persistent and move to support another team, ART, or part of the organization. Or, they may be created for a specific purpose, decommissioned, and returned to their regular work.

Responsibilities and behaviors of enabling teams include:

- Innovative** D identify opportunities for improvement, including adopting new technologies and practices.
- Collaborate proactively** D identify the teams they need to work with, understand their specific requirements, and check progress regularly.
- Communicate regularly** D keep the teams and the broader organization abreast of new technologies and emerging best practices.
- Promote a continuous learning culture** D within their team, the teams they are working with currently, and across the broader organization.

Agile Teams on the ART

In SAFe, teams operate as part of an [Agile Release Train \(ART\)](#). When designing ARTs and the teams composing them, it can be helpful to visualize them considering team topologies. To clarify the team types, we use the icons in Figure 2 to represent them.

These icons can also be used to visualize the likely interactions between the teams through their relative positioning. The names of the specific teams can then be added to these icons for a complete picture. Visualizing the ART's teams helps to compare and contrast the merits of competing designs and indicates how well any particular design is aligned with the flow of value.

Figure 2. Applying team topologies to Agile teams on an ART using Team Topologies at Scale

So far, we have discussed how team topologies can help design the teams that makeup ARTs. But many enterprises also need to organize ARTs that form part of larger [Solution Trains](#). Fortunately, these topologies can be readily extended to help make the right trade-offs in ART design (Figure 3).

Figure 3. A mixture of topologies applied to ARTs within a Solution Train

Figure 3. A mixture of topologies applied to ARTs within a Solution Train

Note: A general exception is the “enabling” team type. Although it is common to have two or three enabling teams working across the enterprise aligned to the same objective, they would unlikely have the scope of an entire ART.

Scaling these topologies to organize ARTs requires additional considerations, as highlighted in the sections below.

Stream-aligned ART

A stream-aligned ART, just like a stream-aligned team, will have the necessary personnel, skills, and authority to deliver value, whether it’s a complete product, service, subsystem, or whatever portion of the solution they have been tasked with.

The areas of responsibility for these stream-aligned ARTs are generally the same as

those for stream-aligned teams. And the same options for aligning them around a particular aspect, as covered earlier, apply here as well.

Complicated subsystem ART

Most large systems also include extensive subsystems. This means complicated subsystem ARTs are expected when building large-scale systems to reduce the cognitive load on the stream-aligned ARTs. For example, an autonomous vehicle's guidance system could require an entire complicated subsystem ART.

Platform ART

Similarly, it's common for a Solution Train to have Platform ARTs providing services that the stream-aligned ARTs extend and build on. Continuing the autonomous vehicle example, a communication system that manages data transferred between the various subsystems would likely be represented as a platform ART with clearly defined interfaces.

One additional benefit of the platform topology is that it supports a single platform ART that provides services across multiple development value streams within the organization, as shown in Figure 4 below.

Figure 4. A platform ART supporting multiple development value streams within a portfolio

The ARTs comprise teams that will take on one of the four-team types in all these examples. For instance, within the complicated subsystem, an ART developing the guidance system may be one or more stream-aligned teams developing the features related to environment perception. Similarly, a complicated subsystem team might focus explicitly on routing algorithms. In this manner, the application of the topologies is fractal.

Of course, there is an intermediate pattern where, within a single ART, there may be a collection of teams working on the same platform or complicated subsystem. In this instance, the work must be carefully allocated to minimize handoffs and dependencies.

Summary

This article brings new patterns to organizing Agile teams and ARTs for large-scale system and software development. Applying the four basic topologies can simplify this complex problem.

Of course, this all relates to the need to continuously reflect on whether our current organizational models serve us well. Thus, organizations must continually Inspect and Adapt (I&A) and, as necessary, reorganize to follow the value driving the market. Organizational Agility demands nothing less.

Appendix: Team Topologies at Scale Example

To illustrate how team topologies can be applied to ARTs and Agile Teams, a financial services example for ABC Bank is provided in this appendix. Specifically, this example will focus on Retail Loans.

ABC Bank has created two Development Value Streams (DVS), which collaborate to support the Retail Loans Operational Value Stream (OVS) shown in Figure 5.

Figure 5. ABC Bank example showing two development value streams

Figure 5. ABC Bank example showing two development value streams

1. **Retail Loans DVS** ð focuses on developing solutions for the origination of retail loans, such as mortgages, personal loans, auto loans, and more.
2. **Core Banking DVS** ð focuses on developing solutions for servicing retail loans and provides other solutions (such as commercial banking, business loans, and investment banking) not described in this example.

Retail Loans DVS

The Retail Loans DVS in Figure 6 includes 340 people (80 in the US, 230 in India, and 30 in Estonia), requiring multiple ARTs. Three options are considered for defining the ARTs teams:

1. Organize ARTs around *channels*: online, mobile, and bank branches
2. Organize ARTs around *customer segments*: existing customers, new customers, students, and homeowners
3. Organize ARTs around *loan products*: mortgages, personal loans, auto loans, and

more.

After analyzing the choices, option three is chosen. Each *loan product* will be an ART within the Retail Loans DVS (as part of a Solution Train), as shown below.

Each ART will serve all customer channels and segments. It's hypothesized that the proposed ART definition will reduce complexity and the number of dependencies, making value flow with less interruption. Another benefit is that this design better aligns with how the bank describes its products and measures business results.

Figure 6. Team topologies applied to the Retail Loans DVS

Figure 6. Team topologies applied to the Retail Loans DVS

The credit scoring subsystem requires highly complex algorithms and artificial intelligence. It requires specialized knowledge and development skills, which are in short supply. After reviewing the design, a decision is made to reduce the cognitive load on the other ARTs in this DVS by making the credit scoring system a complicated subsystem ART. The benefit of this decision is that the specialist will not be distributed across the three ARTs, increasing the credit scoring

subsystem's architectural integrity, especially since AI was recently added to the solution.

The complete decomposition of the Retail Loans DVS is shown in Figure 6.

Core Banking DVS

The Core Banking DVS (Figure 7) serves many bank areas besides retail loans, such as *commercial loans, investment banking, and more*, not shown here. After careful consideration, two ARTs will be created for this DVS:

Figure 7. Core banking DVS and ART Topologies

Figure 7. *Core banking* DVS and ART Topologies

1. **Core Banking Platform ART** provides core banking services for this value stream and others across the broader organization, which is why a dedicated platform ART makes sense.
2. **Retail Loans Stream Aligned ART** develops the retail loans functionality needed to support the Retail Loan OVS.

Organizing Agile Teams within ARTs

The next step is to organize the Agile Teams within each ART illustrated in Figure 8. To simplify the example, we will only decompose one stream-aligned ART (mortgages) for the Retail Loans DVS. The other two stream-aligned ARTs (personal, auto) will use a similar team topology pattern, which is not shown here for simplicity.

Mortgages Stream-Aligned ART

Considering the team structure for the Mortgages stream-aligned ART, the team topologies mentioned earlier in the article are applied, as shown in Figure 8 below.

Figure 8. Team topologies applied to the mortgages stream-aligned ART

Figure 8. Team topologies applied to the mortgages stream-aligned ART

1. ItOs decided that three stream-aligned teams will focus on different *customer segments*: remortgaging, first-time buyers, and existing customers.

2. Another two stream-aligned teams were formed to manage *customer channels*: online channels (for example, web, mobile) and physical channels (for example, bank branches).
3. Finally, two additional stream-aligned teams were formed for internal business processes: compliance and new product innovation.

These stream-aligned teams can deliver value with minimal dependencies on other teams. When they need to collaborate, it's clear which teams must work together as their responsibilities are now well defined.

A loan origination, complicated subsystem team is also formed. Although other teams are capable of interfacing with it via APIs to add and modify the various loan products, it requires deep loan origination knowledge and C programming language skills that are in short supply.

Credit Scoring Complicated Subsystem ART

The credit-scoring complicated subsystem ART is decomposed into four teams, as shown in Figure 9 below.

Figure 9. Team topologies applied to the Credit Scoring Complicated Subsystem ART

It consists of a stream-aligned team, two complicated subsystem teams, and an enabling team:

1. **Credit scoring System** (stream-aligned) ☈ This team works closely with two other complicated subsystem ARTs that must integrate with this system. This example illustrates how the topologies are flexible. In this case, a *stream-aligned team* within a complicated subsystem ART recognizes that this team can deliver value independently for retail loans.
2. **Credit Scoring Algorithm** (complicated subsystem) ☈ designs and maintains highly complex *credit scoring algorithms* for retail loans.
3. **Credit Scoring Exceptions** (complicated subsystem) ☈ creates functionality that authorizes exceptions in the loan application process.
4. **Cloud Technology** (enabling team) ☈ Alongside all this development activity, Retail Loans invests heavily in migrating to the cloud. The credit-scoring complicated subsystem is one of the first systems designated to move to the cloud. In support of this, an *enabling* team will help all teams in this ART migrate to the cloud during the upcoming PI.

The personal and auto loans stream-aligned ARTs must also be decomposed into teams for the Retail Loans DVS (not shown here to keep the example simple).

Finally, the *Core Banking DVS* contains two ARTs described earlier in Figure 7 that needs decomposing into teams and is described in the following sections.

Core Banking Platform ART

The first is a Core Banking Platform ART (Figure 10), which is comprised of four platform teams, four complicated subsystem teams, and an enabling team:

Figure 10. Team structure for the core banking platform ARTs

Figure 10. Team structure for the core banking platform ARTs

1. Four platform teams are formed for different parts of the customer journey: new account creation, account closure, and payment processing (two Agile Teams)
2. Four complicated subsystem teams are formed for shared Core Banking components that are highly complex.
3. An enabling team that supports automated testing will help the core banking

platform ART's teams in the upcoming PI as part of the organization's continuous delivery improvement initiative.

Retail Loans Stream-Aligned ART

The second ART in the Core Banking DVS is *stream-aligned* and focuses on supporting the Retail Loans OVS (Figure 11).

Figure 11. Team structure for the retail loans stream-aligned ART

Figure 11. Team structure for the retail loans stream-aligned ART

1. The Retail Loans stream-aligned ART is decomposed into stream-aligned teams for specific steps in the customer journey: loan default, loan servicing, new loan creation, and loan payments.
2. A complicated subsystem team calculates the loan interest.

The decomposition pattern for this ART works well, allowing teams to deliver value independently and together with limited dependencies.

Learn More

[1] Skelton, Matthew, and Manuel Pais. *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution Press, 2019.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 4 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Break down barriers
between departments.*

NW. Edwards Deming

Organize Around Value

This is article **six** in the **SAFe™ Implementation Roadmap series**. [Click here](#) to view the entire roadmap.

[Adjust Cookie Settings](#)

The first five “critical moves” of the Implementation Roadmap establish the

urgency for change and the critical mass of informed and dedicated people needed to implement SAFe effectively:

- Reaching the Tipping Point
- Train Lean-Agile Change Agents
- Create a Lean-Agile Center of Excellence (LACE)
- Train Executives, Managers, and Leaders
- Lead in the Digital Age

With a sense of urgency and a powerful coalition of empowered change agents in place, it's now time to implement SAFe. This article describes the next critical move—[Organize Around Value](#). Value streams and ARTs are the organizational backbone of a SAFe implementation and are critically important to the success of this journey. Attempting to shortcut or breeze through this step would be like putting your foot on the brake while simultaneously trying to accelerate. But get this one right, and the organization will be well on its way to a successful transformation.

Details

Organizing around value requires an understanding of a new organizational model, one that is optimized to facilitate the flow of value *across* functional silos, activities, and boundaries and includes the following steps:

- Identify the Operational Value Streams
- Identify the Solutions the operational value streams use or provide to customers
- Identify the people who develop and support the solutions
- Identify the Development Value Streams that build the solutions
- Add the people needed to make the whole business solution
- Realize development value streams into ARTs

The sections below describe each of these activities.

A Value Stream Refresher

A value stream is the primary construct for understanding, organizing, and delivering value in SAFe. As illustrated in Figure 1, each value stream is a long-lived series of steps used to create value—from concept to the delivery of a tangible result for the customer. Like any well-constructed narrative, the value stream identifies a chronological flow of activities:

Figure 1. The anatomy of a Value Stream

Figure 1. The anatomy of a Value Stream

- Trigger** ◊ Some important event triggers the flow of value, perhaps a customer purchase order or new feature request. It ends when some value—*a shipment, customer purchase, or solution deployment*—has been delivered.
- Steps** ◊ The chevrons in the middle are the steps the enterprise uses to accomplish this feat [1].
- Value** ◊ The customer receives value when the value stream executes its steps.
- People and systems** ◊ A value stream also contains the people who do the work, the systems they operate, and the flow of information and materials.
- Lead time** ◊ The time from the trigger to the value delivery is the total lead time. Shortening the lead time reduces the time to market. The easiest way to shorten lead time is to identify and reduce (or remove) non-value-added activities and wasteful delays. That's the primary focus of Lean thinking.

Types of Value Streams

Note that there are two types of value streams [1], as illustrated in Figure 2.

Figure 2. Operational and development value streams

Figure 2. Operational and development value streams

Operational Value Streams Ⓛ Contain the steps and the *people* who deliver end-user value using the business solutions created by the development value streams.

Development Value Streams Ⓛ Contains the steps and the *people who develop* the business solutions used by *operational value streams*. (These are the value streams in a SAFe portfolio.)

Development value streams are SAFe's primary concerns. After all, delivering new solutions in the shortest sustainable lead time is the focus of SAFe, and development value streams help organizations understand how to get there. However, the enterprise's operational value streams must be identified first to determine the *development value streams* that support them.

Identify Operational Value Streams

For some organizations, identifying operational value streams is easy. Many are just the steps to produce the products, services, or solutions that the company sells. In

the larger enterprise, however, the task is more complicated. Value flows through various applications, systems, and servicesÑacross many parts of the distributed organizationÑto *internal* and *external* customers.

Identifying operational value streams in a large enterprise is not a trivial undertaking. It requires an awareness of the organizationÕs broader purpose and an explicit understanding of how specific elements of value flow to the customer. The below questions can help stakeholders through that identification process.

Who are your users and customers? Are they internal, external, or both?

What products and services do you market, sell, and support?

What solutions do you provide for internal users?

What triggers the flow of value?

What value do your customers perceive you deliver?

Most generally, operational value streams fall into one of four types shown in Figure 3:

1. *Fulfillment value streams* represent the steps necessary to process a customer request, deliver a digitally-enabled product or service, and receive remuneration. Examples include providing a consumer with an insurance product or fulfilling an eCommerce sales order.
2. *Manufacturing value streams* convert raw materials into the products customers purchase. Examples include consumer products, medical devices, and complex cyber-physical systems.
3. *Software product value streams* offer and support software products. Examples include ERP systems, SaaS, and desktop and mobile applications.
4. *Supporting value streams* include end-to-end workflows for various supporting activities. Examples include employee hiring and retention lifecycle, supplier contracting, executing the annual budget process, and completing an entire enterprise sales cycle.

Figure 3. Examples of the four types of operational value streams

Figure 3. Examples of the four types of operational value streams

And although companies need strong, functional departments to build and share knowledge, sales, marketing, purchasing, legal, finance, and manufacturing engineering are not value streams. However, many of the people in these departments participate in one or more operational value streams.

It stands to reason that large enterprises typically offer their customers a wide variety of products and services. That's one of the ways they grow. As such, it follows that there are substantial value streams within those enterprises. Figure 4 illustrates how the *Consumer loan* value stream is just one of a large commercial bank's offerings and operational value streams [2].

Figure 4. Operational value streams in a commercial bank

Figure 4. Operational value streams in a commercial bank

Value Stream Definition Template

The value stream definition template in Figure 5 can be used as an aid to elaborate and understand the characteristics of the identified operational value stream.

Figure 5. Value stream definition template with an operational value stream example

Identify the Solutions the Operational Value Stream use or Provide to Customers

Once the Operational Value Stream steps are identified, the next activity is identifying the solutions needed to support it. It's essential to map the connections from the solutions to the various steps in the value stream and identify the people who operate these solutions. The map creates a deeper understanding of how it all works, as the consumer loan example illustrates in Figure 6.

Figure 6. Identifying the systems that support the steps

Figure 6. Identifying the systems that support the steps

Identify the People who Develop and Support the Solutions

Once the solutions that support the operational value stream have been identified, the next activity is to estimate the number and locations of the people that build and maintain those solutions, as Figure 7 illustrates.

Figure 7. Identifying the people who develop the systems

Figure 7. Identifying the people who develop the systems

Identify the Development Value Streams that Build the Solutions

The next step is to identify the development value streams, which represent the steps needed to develop those solutions and the people who create them. The solutions support and enable better operational value streams; as such, the value is new or amended features for the solutions. The triggers for the development value streams are the needs and ideas that drive these features.

Figure 8. Defining the development value streams

Figure 8. Defining the development value streams

These triggers are also used to identify the development value streams. If most requirements necessitate touching all solutions to enable the new functionality, there is likely only one development value stream. If the solutions are decoupled, there might be a few of them. In any case, development value streams should be mostly or wholly independent and able to develop and release by themselves without too many intra-value stream dependencies. In the example in Figure 8, most requirements touch the first three systems or the last one, but rarely all four, so there are two development value streams, each capable of developing, integrating, deploying, and releasing independently of the other.

While there is no apparent limit or constraint to ways an enterprise can configure development value streams, specific patterns have emerged. These are discussed in detail in the [Development Value Streams](#) article.

Add the People Needed to Build the Full

Business Solution

Development value streams strive to deliver innovative business solutions and, as such, require the contributions of more than just Agile teams. Everyone involved in business solution development – including representatives of IT operations, legal, marketing, finance, support, compliance, security, and others – is considered part of the development value stream. With this in mind, the next step is to identify these additional individuals and teams who are part of the development value streams identified in the previous step.

Figure 9. Add the People Needed to Build the Full Business Solution

Figure 9. Add the People Needed to Build the Full Business Solution

In the example in Figure 9, the first development value stream, focused on the loan application process, includes marketing to develop promotional materials and campaigns for attracting customers. Legal team members are also included to define the relevant terms and conditions of the loans offered. The second development value stream, which develops the Core Banking system that manages the repayment of loans, also includes marketing for managing ongoing

communications with the existing customer base. Both development value streams include support teams to respond to any customer issues that might arise and operations teams that manage the stability of these systems in production.

Development Value Streams Cross Boundaries

Once the development value streams are identified, the next step is understanding how to form Agile Release Trains(ARTs) to realize them. The ARTs contain all the people and other assets needed to enhance the *flow* of value. The first step is to understand where value is created in the organization because that is where the people, processes, and systems are. It becomes evident that development value streams cross many boundaries when doing so. Enterprises are organized the way they are for many reasons: history, functional convenience, the efficiency of centralization, acquisitions, geography, and more. As a result, no one may understand the complete series of events necessary to develop and enhance the systems that help deliver value continually. Furthermore, improvement attempts often focus on functional, local improvements, which may result in the optimization of one function or step but sub-optimization of the end-to-end flow.

The long-lived nature of value streams triggers different thinking in the Lean organization. Typically, larger enterprises are organized functionally. In addition, people are often distributed geographically. But value moves *across* these boundaries, as Figure 10 illustrates.



Figure 10. Value flows across functional, organizational, and geographic boundaries

Realize Development Value Streams into ARTs

The final activity is to define the ARTs that realize the value. Experience has shown that the most effective ARTs have the following attributes:

50 ÷ 125 people

Focused on a holistic solution or related set of products or services

Long-lived, stable teams that consistently deliver value

Minimize dependencies with other ARTs

Can **release** independent of other ARTs

Depending on how many people do the work, there are three possible scenarios for the ART design, as Figure 11 illustrates.

Figure 11. Three possible scenarios for ART design

Figure 11. Three possible scenarios for ART design

Multiple development value streams can fit within a single ART Ⓛ When several related products or solutions can be produced with a relatively small number of people, a single ART may deliver multiple value streams.

A single development value stream can fit within an ART Ⓛ Often, a Value Stream can be realized with 100 or fewer practitioners. Many development groups are already organized into units of about that size, so this is a typical case. In this case, the ART is roughly the same as the value stream. Everyone is in that ART!

Multiple ARTs are required for large development value streams Ⓛ When many people are involved, the development value stream must be split into multiple ARTs, as described in the next section, and form a Solution Train.

Splitting Large Value Streams into Multiple ARTs

Large development value streams are ubiquitous in large enterprises, and additional analysis is often required. Trains should focus on a single primary solution or a set of closely related products or services. In smaller value streams, the result is a relatively simple design—none ART delivering a well-defined set of

solutions.

However, when many people are needed to deliver a single solution, multiple ARTs will need to collaborate as part of a Solution Train. To support the overall goal of continuous value delivery, each ART within the Solution Train must be designed to maximize flow across the entire Solution Train.

As described in the [Agile teams](#) and [Agile Release Train](#) articles, SAFe recognizes fundamental [team topologies](#) [4] to help with the job of team and ART design, which are defined as follows:

1. **Stream-aligned team** Ø organized around the flow of work and has the ability to deliver value directly to the customer or end-user
2. **Complicated subsystem team** Ø organized around specific subsystems that require deep specialist skills and expertise
3. **Platform team** Ø organized around the development and support of platforms that provide services to other teams
4. **Enabling team** Ø organized to assist other teams with specialized capabilities and help them become proficient in new technologies

These topologies can help make the right trade-offs in ART design in the context of a [Solution Train](#) (Figure 12).

Figure 12. A mixture of topologies applied to ARTs within a Solution Train

Figure 12. A mixture of topologies applied to ARTs within a Solution Train

(Note: A possible exception when applying these topologies to ARTs is the “enabling” team type. Although it is common to have two or three enabling teams working across the portfolio aligned to the same objective, it is unlikely this would represent an entire ART on a Solution Train.)

Scaling these topologies to organize ARTs requires additional considerations, as highlighted in the sections below.

Stream-aligned ARTs

A stream-aligned ART, just like a stream-aligned team, will have the necessary personnel, skills, and authority to deliver value, whether it’s a complete product, service, subsystem, or whatever portion of the solution they have been tasked with.

The areas of responsibility for these stream-aligned ARTs are generally the same for stream-aligned teams. And the same options for aligning them around a particular aspect, as covered earlier, apply here as well.

Complicated subsystem ART

Most large systems also include extensive subsystems. This means complicated subsystem ARTs are expected when building large-scale systems to reduce the cognitive load on the stream-aligned ARTs. For example, a guidance system for an autonomous vehicle could well require an entire complicated subsystem ART.

Platform ARTs

Similarly, it's common for a Solution Train to have Platform ARTs providing services that the stream-aligned ARTs extend and build on. Continuing the autonomous vehicle example, a communication system that manages data transferred between the various subsystems would likely be represented as a platform ART with clearly defined interfaces.

(Note: One additional benefit of the platform topology is that it also supports a single platform ART that provides services across multiple development value streams within the organization.)

And, of course, combinations of these types often appear in larger value streams, as our final example in Figure 13 illustrates.

Figure 13. Combinations of ART topologies in the consumer bank loan example

Figure 13. Combinations of ART topologies in the consumer bank loan example

Finally, other ART designs and optimization factors are based on concerns such as geography, spoken language, and cost centers, which may influence the ART design. But these are far less desirable.

The SAFe Value Stream and ART identification Workshop

As demonstrated, there's critical thinking and analysis involved in this process. To help identify value streams, Scaled Agile, Inc. provides a [Value Stream and ART identification workshop toolkit](#), consisting of a workshop and other artifacts that [SAFe Practice Consultants \(SPCs\)](#) can use to guide stakeholders. The workshop provides a structured approach to identifying value streams and defining ARTs, which can realize the flow of value in the

enterprise. This toolkit offers a proven, systematic approach to optimizing ART design by considering the dependencies, coordination, and constraints.

The Value Stream and ART identification workshop is often run directly following a [Leading SAFe](#) class with critical stakeholders. The objective is to take them through identifying the value streams, designing the ARTs, and perhaps even picking the date for the first ART launch *after* they have a fundamental understanding of Lean-Agile development enabled by SAFe.

Because no design is perfect, enterprises sometimes repeat this workshop after learning more as part of the [Accelerate](#) roadmap step. Doing this allows enterprises to refine their understanding of value streams and ARTs and incorporate new learnings into the organizational design.

Moving Forward

This article described how teams do the work to organize around value and design the ARTs that form the basic organizational structure for the transformation.

Some change leaders may find identifying their portfolios and beginning the initial steps of organizing the [Portfolio](#) related to the initial ARTs identified assists in this acceleration. In this case, activities described within [Enhance the Portfolio](#) may be begun here, with the knowledge that these critical moves are not dependent on each other but have the potential of accelerating each other early on.

Now it's time for the next step, [Create the Implementation Plan](#), which is the next article in the SAFe Implementation Roadmap.

NEXT

Learn More

[1] Ward, Allen. *Lean Product and Process Development* (video). Lean Enterprise Institute, 2004.

[2] Contributed by Wilmshurst, Darren, Murray Ford, Per-Magnus Skoogh, Phillip Manketo, Sam Bunting, and Virpi Rowe.

[3] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[4] Skelton, Matthew, and Manuel Pais. *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution Press, 2019.

Additional Resources

SPCs have access to a Value Stream and ART Identification Workshop explicitly designed to help identify value streams, initially define ARTs, and create the implementation plan. [Learn more about the toolkit.](#)

Last update: 19 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

TRAINING

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS

[YouTube](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Course Calendar](#)

[Privacy Policy](#)

[About Certification](#)

[Cookie Policy](#)

[Become a Trainer](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The objective of the pull event was simple. It was designed to focus the development organization on a tangible event to force completion of a learning cycle with the objective to physically demonstrate it.

~Dantar P. Oosterwal

Adjust Cookie Settings

Solution Demo

The Solution Demo provides stakeholders an integrated view of the contributions of multiple ARTs and suppliers to obtain objective evidence of solution performance and to gather feedback.

The Solution Demo presents the combined development efforts of multiple [Agile Release Trains \(ARTs\)](#) along with the contributions of [Suppliers](#) and other solution participants to [Customers](#) and other stakeholders. This demo is critical for the [Solution Train](#) to receive objective evaluation and feedback. It's also a moment to celebrate the accomplishments of the last PI.

Each solution demo represents a significant learning point in the history of the [Solution](#), converting some product development uncertainty into knowledge. The results of this demo determine the future course of investment in the solution.

Details

During the solution demo, [Agile Teams](#) demonstrate the solution's new [Capabilities](#), its compliance with [Nonfunctional Requirements](#) (NFRs), and its overall fitness for purpose. To ensure progress throughout the PI, ARTs and suppliers should strive to continuously or partially integrate their changes whenever possible. (See also the discussion of frequent integration in [Enterprise Solution Delivery](#)). At a minimum, changes across the entire Solution Train should be integrated before the solution demo. (Figure 1).

Figure 1. Solution Trains integrate changes at least every PI

The solution demo provides essential input to near-term [Development Value Stream](#) and [Portfolio Level](#) investment decisions. The objective measure of progress gives early validation and mitigates investment risk.

Solution Demo as a “Pull” Event

As the quote at the top of this article suggests, the solution demo is a deliberate and high-profile “pull” event. It “pulls” together various aspects of the solution and helps ensure that the ARTs and suppliers create an integrated and tested solution that is fit for its intended purpose. This demo accelerates the integration, testing, and evaluation of the entire solution under development—something that’s all too easy to defer until too late in the development lifecycle.

[Enterprises](#) sometimes create even more significant pull events within a portfolio, during which several Solution Trains come together for a “roadshow” of their accomplishments (see [1] for example). The portfolio’s senior leaders and stakeholders review progress in a broader context and make decisions about initiatives’ continuation (or cancellation). Or, they might decide to make changes to the investment in their development value streams.

Preparation

A solution demo is a critical event that requires preparation, which often begins with [Pre-Plan](#), where the results of the most recent solution demos are available. Those results inform the people staging the demo on what specific capabilities and aspects of the solution can and should be demonstrated. Since the Solution Train is large and usually distributed, logistics do matter. Video conferencing technology should support remote attendees to observe and participate. Timing, presentation format, and professionalism enhance the experience. It may even influence the outcome.

Attendees

Attendees for a solution demo typically include:

- [Solution Management](#)
- [Solution Train Engineers \(STEs\)](#)
- [System and Solution Architects](#)
- Customers

ART representatives, [Product Management](#), the [System Team](#), [Product Owners](#), and representatives of the Agile teams themselves (to experience the customer feedback firsthand)

[Lean Portfolio Management \(LPM\)](#) representatives

Solution Train stakeholders, executive sponsors, and senior leaders

Deployment operations representatives

Event Agenda

A typical event agenda includes the following activities:

Briefly review the Solution Train [PI Objectives](#) for the PI

Demo each PI objective and capability in an end-to-end use case

Identify business value completed for each PI objective

Open forum for questions and comments

Wrap up by summarizing progress, feedback, and action items

The day can be even more interesting if multiple solutions are demoed together. A popular format is the “science fair,” where each solution has an area to demo progress and allow stakeholders to ask questions and provide feedback. Each solution has a timebox to demo its accomplishments to a set of specific stakeholders following the agenda above, but all solutions are constantly available for demo. Members from other solutions and stakeholders can attend each other’s demos to get a less formal demonstration and provide feedback.

Guidelines

Here are a few tips to keep in mind for a successful demo:

Timebox the demo to one to two hours. Sticking to the allotted timebox demonstrates professionalism, respect for people’s time, and solution readiness. Share demo responsibilities among lead engineers and team members who have new capabilities to demo.

Minimize slides; demonstrate working and tested capabilities only.

Minimize demo preparation. Demo the working, tested capabilities, not slideware.

Discuss the impact of the current PI on the solution NFRs and [Solution Intent](#).

Demonstrate in the [Solution Context](#) (see below).

Demonstrate the Solution in Its Solution Context

Solutions may have varying degrees of coupling with their solution context. Sometimes, a solution is mainly independent of its environment, and an isolated solution demo may be adequate. However, when the system is highly dependent on the solution context (a system of systems, for example), the isolated approach is inadequate and may even be misleading. In this case, the solution should be demoed in an environment that is representative of its solution context. When that's not practical, development should plan for some cadence of integration with the broader solution context.

Strategy, Investment, and Timing of Solution Demos

Large systems can be hard to integrate. The [Solution Roadmap](#) communicates the anticipated, demonstrable capabilities for upcoming PIs that align ARTs and suppliers and set expectations for what can be demonstrated at any point in time. Routinely demonstrating a solution increment requires teams to invest in [the CD Pipeline](#), grow testing and [Built-in Quality](#) practices, and evolve supporting infrastructure. Even then, the extent of integration and testing may be less than 100 percent and may need to vary across multiple integration points. To reduce the integration burden, teams leverage virtualization, environment emulation, mocks, stubs, and reduced test suites to demo the completed parts. Also, some effort for integration, demo, and time to invest in the supporting environment may need capacity allocation during [PI Planning](#).

As for timing, the solution demo may lag slightly behind the last system demos in the PI. However, this creates a delayed feedback loop that increases risk and decreases Solution Train velocity. Here are some tips for minimizing the lag:

Plan to demonstrate a subset of the PI objectives, which may require some staging and configuration management to support partial demos.

Set aside time during the [Innovation and Planning \(IP\) Iteration](#) for this high-level integration.

ARTs that broaden their areas of responsibility for integration and testing can create more overlap with the subsystems and capability areas of other trains. As a result, even the individual system demos offer a better approximation of the fully integrated solution demo.

Finally, the solution may be designed to support better integration and testing, significantly lowering the demo's cost. [Agile architecture](#) practices such as standard interfaces, strictly defined APIs, and containers help teams spot problems and inconsistencies early on, making end-to-end integration and testing of subsystems

easier.

Learn More

[1] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. Amacom, 2010.

Last update: 1 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)
[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL
[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)

English

Scaled Agile Framework



Stay committed to your decisions but stay flexible in your approach.

—Tom Robbins

Iteration Planning

Note: For more on SAFe Scrum, please read the additional Framework articles in the Scrum series, including [SAFe Scrum](#), [SAFe Scrum Master/Team Coach](#), [Iterations](#), [Iteration Goals](#), [Iteration Review](#), and [Iteration Retrospective](#)

Iteration planning is a SAFe Scrum event where all team members determine how much of the Team Backlog they can commit to delivering during an upcoming Iteration. The team summarizes this work as a set of committed iteration goals.

Details

Iteration planning is the first event of the [Iteration](#). During planning, the team defines, organizes, and commits to the work for the next iteration. The iteration planning meeting is timeboxed to a maximum of 90 minutes for a two-week iteration. The team's backlog has been partially identified and planned during [PI Planning](#). In addition, the teams have feedback—not only from their prior iterations but also from the [System Demo](#), stakeholders, and others. All this context feeds into the iteration planning event to inform the plan for the upcoming iteration.

Inputs and Outputs of Iteration Planning

Inputs to iteration planning include:

- A refined [Team Backlog](#), including [Stories](#) from the team's PI plan, which were identified during [PI Planning](#) and tentatively assigned to iterations, new stories that have been identified, and other items from the team's local context, including defects, refactors, maintenance, and technical debt
- The Team and ART [PI Objectives](#) created during PI planning
- Feedback from the [System Demos](#) and prior iterations, including any stories that did not meet the definition of done (DoD)

A successful iteration planning event delivers the following outputs:

- Stories planned for the upcoming iteration, including Enablers. Each has defined acceptance criteria and an estimate and is recorded in the iteration backlog.
- Committed iteration goals.
- Dependencies with other teams are understood and planned.

Preparation

The teams prepare for the event as follows:

- Backlog refinement.** Teams usually approach iteration planning with a pre-elaborated team backlog from the backlog refinement sessions held during the

previous iteration.

Close out the previous iteration. The team confirms the stories in the last iteration were completed and accepted. If any remain, they are moved to the team backlog and reprioritized.

Initial iteration goals. The Product Owner (PO) may prepare some initial iteration goals based on the team's progress in the PI.

Process

The PO typically starts the event by presenting high-priority stories from the team backlog and the initial iteration goals (if applicable). Many of these stories originated from PI planning, while others are from the team's local context. The acceptance criteria are then elaborated through conversation and collaboration with the Product Owner and other stakeholders. Next, they estimate the effort to complete each item using relative story points. Based on the estimates and business value, the Product Owner may change the ordering of the stories.

During planning, the team discusses implementation options, technical issues, [Nonfunctional Requirements \(NFRs\)](#), and dependencies. Next, the PO and team select the candidate stories based on their available capacity for the iteration. Some teams decompose stories into tasks (optional) and forecast them in hours to confirm they have the capacity and skills to complete them. Once planning is complete, the team synthesizes the work into iteration goals, commits to the plan, and records the iteration backlog in a visible place, such as a story or Kanban board and Agile project management tooling.

Attendees

Attendees of the iteration planning event include:

The [Product Owner](#)

[Scrum Master/Team Coach](#)

All team members and any subject matter experts, as needed

Any other stakeholders required, including representatives from other Agile Teams or trains

Scrum Masters/Team Coaches or POs typically facilitate iteration planning for the team, ensuring the participants stay within the agreed agenda and event timebox.

Agenda

An example agenda for iteration planning follows (Figure 1), including a description of each item.

Figure 1. Example iteration planning agenda

Figure 1. Example iteration planning agenda

During planning, the PO defines the “what,” the team determines the “how,” and “how much,” as follows:

1. **Establish capacity** The team calculates its capacity for the upcoming iteration using adjusted historical velocity.
2. **Story analysis and estimation** In conversation with the PO, the team selects and estimates the most valuable stories to meet their PI Objectives, addressing local concerns and dependencies, where applicable.
3. **Tasking stories (optional)** Some teams task stories to understand their capacity and capabilities better. They determine the best person to accomplish the work, estimate the effort (typically in hours), and identify dependencies with other tasks or stories. Planning stops once the team runs out of capacity.
4. **Develop iteration Goals** This process repeats until when the team is out of capacity. Next, the teams summarize the plan as a set of iteration goals. (Note: Some teams work the other way around; they start with iteration goals and then work on capacity, story analysis, and estimating to support those goals.)
5. **Commit to iteration goals** At the end of planning, the Product Owner and team agree on the final list of stories that will be selected, and they revisit and restate the iteration goals. Everyone commits to the iteration goals, and the scope of the work remains largely fixed for the duration of the iteration.

Commitment to Iteration Goals

Iteration goals provide clarity, commitment, and information. The commitment to the iteration goals is reciprocal (Figure 2) and serves the following purposes:

- Aligns team members to a set of shared objectives for the iteration
- Focuses teams on meeting their PI objectives and managing dependencies with

other teams

Provides transparency and management information as needed

Figure 2. Guidelines for team commitments

Figure 2. Guidelines for team commitments

Estimating Stories and Forecasting Team Capacity

Relative Estimation of Stories for Planning

Agile Teams use story points to estimate stories relative to each other [2, 3]. The size (effort) for each item is compared to other stories. For example, an eight-point story should be four times the effort of a two-point story. (Note: Refer to the [Story](#) article to learn how to write and estimate stories, including practices for whole team estimation and how to split large stories so they can be completed within an iteration.)

Forecasting Team Capacity

The team forecasts its *capacity* for an upcoming iteration by using its historical velocity as a starting point. The team's average velocity (completed story points per iteration) becomes more reliable and predictable as they work together. Predictable velocity helps with planning and limits Work in Process (WIP).

Starting Capacity for New Teams

When the team is new and the average velocity is *unknown*, one method for initially forecasting the team's capacity is as follows:

1. Give the team 8 points for every full-time developer (including test, etc)
2. Subtract one point for every team member's vacation day, holiday, or other non-working days in the iteration.

Figure 3 provides an example of forecasting capacity for a new seven-person team.

Figure 3. Example starting capacity for a new team

Creating a Shared Basis for Story Point Estimation

Story points can be aligned (approximately normalized) by teams in the ART, providing a *shared basis* for capacity forecasting and economic decision-making.

One approach is as follows:

1. Each team finds a small story that would take a day to develop, test, and validate. Call it a Ÿone.Ó
2. Estimate every other story relative to that Ÿone.Ó

Moreover, aligning story points enables reasonable estimate costs for features and epics requiring multiple teamsÓ support. While many teams will increase their velocity over timeÑand thatÑs a good thingÑ in reality, the number tends to remain stable.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[3] Cohn, Mike. *Agile Estimating and Planning*. Robert C. Martin Series. Prentice-Hall, 2005.

Last Update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and](#)

SCALED AGILE, INC

[CONTACT US](#)

Watch and download
SAFe videos and
presentations

Blog

TRAINING

Course Calendar

About Certification

Become a Trainer

GET SOCIAL

Twitter

Linkedin

YouTube

trademarks

Permissions Form

Usage and Permissions

5400 Airport Blvd., Suite

300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Set a goal and don't quit until you attain it. When you do attain it, set another goal, and don't quit until you reach it. Never quit.

—Bear Bryant,
University Alabama
football coach

Iteration Goals

[Adjust Cookie Settings](#)

Note: For more on SAFe Scrum, please read the additional Framework articles in the Scrum series, including [Scrum Master/Team Coach](#), [Iterations](#), [Iteration Planning](#), [Iteration Goals](#), [Iteration Review](#), and [Iteration Retrospective](#)

Iteration Goals are a high-level summary of the business and technical goals that an Agile Team agrees to accomplish in an Iteration.

Iteration goals provide the following benefits:

- Align team members to a common purpose
- Align teams to common [PI Objectives](#) and manage dependencies
- Provide transparency and management information

Iteration goals provide [Agile Teams](#), [Agile Release Train \(ART\)](#) stakeholders, and management with a shared language for maintaining alignment, managing dependencies, and making necessary adjustments during the execution of the [Planning Interval](#). These goals apply to teams whether they use [SAFe Scrum](#), [SAFe Team Kanban](#), or a hybrid of both.

Details

As described in the [Iteration Planning](#) article, the planning process produces three primary outputs:

- Stories planned for the upcoming iteration, including [Enablers](#). Each item has defined acceptance criteria and an estimate, which the Team records in their [Team Backlog](#).
- A set of committed iteration goals (Figure 1).
- Dependencies with other teams are understood and planned.

Figure 1. One team's iteration goals

Figure 1. One team's iteration goals

Iteration goals often reflect the following:

- Aspects of the team and ART PI objectives
- [Features](#), slices of feature slices, or other aspects
- Business or technical milestones (see [Roadmaps](#))
- Architecture, infrastructure, exploration, and compliance activities
- Routine jobs and other things, such as maintenance, refactoring, and documentation

Teams achieve iteration goals by completing backlog items, even though finishing every story or enabler may not be necessary to meet the objectives. In other words, achieving the goals for the iteration are more important than completing a particular backlog item. Adding new stories to meet the iteration's objectives may also be required. Simply put, the iteration goal provides flexibility regarding the work needed to achieve it.

Why Iteration Goals?

Iteration goals support two of SAFe's four [Core Values](#), alignment and transparency. Simply committing to complete a set of stories in an iteration is insufficient. The team must continually review the business value of each iteration and then be able to communicate it in business terms to the [Business Owners](#), management, and other stakeholders. The iteration goals also create coherence and focus, encouraging the team to work together rather than on separate stories, which improves flow by limiting work in process (WIP).

In the ART context, iteration goals help create alignment, understand, and maintain a larger view of what the team intends to accomplish in each iteration and what to present in the upcoming [System Demo](#).

Although SAFe Kanban teams don't typically use iterations for planning purposes in the same way that SAFe Scrum teams do, iteration goals provide the necessary transparency and alignment.

Align Team Members to a Common Purpose

The execution of an iteration goes by very quickly. It's a fast and furious process. Iteration goals help the team and [Product Owner](#) to agree on the business value they intend to deliver, align their work to their team PI objectives, and ground everyone on their shared purpose, as Figure 2 illustrates.

Figure 2. Iteration goals help align the team

Figure 2. Iteration goals help align the team

Align Teams to Common ART PI Objectives and Manage Dependencies

Agile teams are integral parts of the broader ART context and mission. As a result, the intent of upcoming iterations requires communication with other teams and

the [Release Train Engineer \(RTE\)](#). Iteration goals facilitate alignment with the ART PI Objectives. Also, they provide the necessary context for discovering dependencies and developing a resolution, as Figure 3 illustrates.

Figure 3. Iteration goals align teams to PI objectives and each other and help identify dependencies

Provide Transparency and Management Information

Scaling agility depends on creating a leaner, more empowered organization where management can decentralize responsibility and focus instead on eliminating impediments and driving improvements.

However, management cannot and should not relinquish its responsibility to understand what the teams are doing and why they are doing it. Managers are still

accountable for the effectiveness of the development organization and the value delivery outcomes. Aggregating iteration goals for a train provides a simple, transparent, two-week summary of what's happening, as Figure 4 illustrates.

Figure 4. Iteration goals provide visibility and communication with management

Figure 4. Iteration goals provide visibility and communication with management

Whether Agile Teams apply Scrum or Kanban, iteration goals provide Agile teams, Agile Release Train (ART) stakeholders, and management with a shared language for maintaining alignment, managing dependencies, and making necessary adjustments during the execution of the PI.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled: Achieving Business*

Last update: 11 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

- Weekdays: 9am to 5pm
- Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Few are those who see with their own eyes and feel with their own hearts.

ÑAlbert Einstein

Iteration Review

Note: For more on SAFe Scrum, please read the additional Framework articles in the Scrum series, including [SAFe Scrum](#), [Scrum Master/Team Coach](#), [Iterations](#), [Iteration Planning](#), [Iteration Goals](#), and [Iteration Adjust Cookie Settings](#)

Retrospective

The Iteration Review is a regular SAFe Scrum event where the team inspects the iteration increment, assesses progress, and adjusts the team backlog.

Details

The iteration review is the second to last event of the iteration. It provides a way to regularly gather immediate, contextual feedback from the team and its stakeholders. The iteration review offers several benefits:

- It brings closure to the iteration timebox
- It allows team members to demonstrate their contributions and to take some satisfaction and pride in their work
- It provides an opportunity for the team to receive feedback to improve the solution under development
- It shows the results of the latest system increment to help determine future work

An iteration review is where the team demos a working, tested increment. No slides are needed. Instead, the focus is on the solution instead of a presentation. The team and stakeholders review the accomplishments in the iterationÑbased on this information, attendees collaborate on what to do next. The [Team Backlog](#) may also be adjusted to meet new opportunities.

Inputs and Outputs of the Iteration Review

Inputs to the iteration review include:

- Iteration goals and [PI Objectives](#)
- The teamÑs increment deployed to a staging environment (or production environment where appropriate)
- A brief list of work to be demoed to prepare people for what they are about to see

A successful iteration review event delivers the following *outputs*:

- Feedback on the increment and progress toward the iteration goals and broader PI Objectives
- Adjusted Team Backlog based on feedback
- Identification of risks and impediments

Preparation

The preparation for the iteration review begins during Iteration Planning, where teams start thinking about how they will demo the committed [Stories](#). Beginning with the end in mind facilitates iteration planning and alignment, fostering a more thorough understanding of the functionality needed ahead of iteration execution.

Process

The PO starts the iteration review by discussing the iteration goals and their status. It proceeds with a walk-through of all the committed stories. Teams demonstrate the significant new behavior and knowledge gained from the iteration's completed stories, [Spikes](#), [Refactors](#), and [Nonfunctional Requirements \(NFRs\)](#). The demos should be part of a working, tested system—preferably in a staging environment closely resembling production. Spikes and NFRs can be demonstrated via a presentation of findings if the functionality lacks a user interface. Stakeholders provide feedback on the stories that the team demoed, which is the primary goal of the review.

The team reflects on stories *not* completed after the demo and why they could not finish them. This discussion usually results in discovering impediments or risks, false assumptions, changing priorities, estimating inaccuracies, over-commitment, or other problems with [Team Flow](#). These findings often lead to further study in the [Iteration Retrospective](#) and may result in improvements to support better planning and execution going forward. Figure 1 shows an iteration review in action.

Figure 1. An Agile team demoing a working, tested increment

The team reflects on how well it did within the iteration and determines its progress toward its Team PI objectives. It finishes the event by refining the Team

Backlog, based on the feedback received, before the next iteration planning event.

Attendees

Attendees at the iteration review include:

The Product Owner (PO)

Scrum Master/Team Coach

All team members and other stakeholders or subject matter experts

Stakeholders, which may also include other teams or trains.

Although [Agile Release Train \(ART\)](#) stakeholders may attend, their interests and the level of detail they require are usually better aligned with the [System Demo](#).

Agenda

The timebox for the event is a maximum of 90 minutes for a two-week iteration.

Figure 2 shows an example agenda and a description of each item.

Figure 2. An example iteration review agenda

The Scrum Master/Team Coach or PO typically facilitates the iteration review for the team, ensuring they stay within the agreed event agenda and timebox.

Following are descriptions of the example agenda:

1. **Review iteration goals** □ Discuss the status of each iteration goal. Teams may also review PI objectives for a broader context.
2. **Demonstrate completed stories** □ The iteration review proceeds with a walk-through and demonstration of each completed story (spikes, NFRs, and any other work finished by the team). Demos should show progress towards iteration goals, PI

objectives, solution changes, test scenarios, or a prototype representing the user's environment. Spikes can be demoed as a presentation of findings or learning. The team and stakeholders should ask questions and provide constructive feedback.

3. **Reflect on any incomplete stories** ▷ Next, the team should reflect on missed iteration goals and stories they did not complete to identify opportunities for future improvement. This discussion usually results in discovering impediments or risks, false assumptions, changing priorities, estimating inaccuracies, or over-commitment.
4. **Identify risks and impediments** ▷ After the demo and reflecting on any incomplete stories, the team identifies new risks or dependencies that might impact achieving the PI objectives. Teams often use the ROAM (Resolved, Owned, Accepted, Mitigated) process to address the risks as needed.
5. **Refine the Team Backlog** ▷ Based on stakeholder feedback, the team can refine their backlog to reflect any adjustments before the next [Iteration Planning](#) event.

Guidelines

Below are some tips for running a successful iteration review event:

- Limit preparation to less than two hours
- Timebox the event to about 90 minutes
- Minimize the use of slides; the iteration review is intended to garner feedback on working, tested system components
- Verify that completed stories meet the definition of done (DoD)
- Demonstrate incomplete stories if enough functionality is available to get feedback
- Encourage providing constructive feedback and celebration of the team's accomplishments
- If a significant stakeholder cannot attend, the PO should follow up to report progress and get feedback

Note: Teams applying [Continuous Delivery](#) will generally review completed Stories or [Features](#) as soon as they are available rather than waiting until the end of the iteration.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.

Last update: 23 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

[Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

~Agile Manifesto

Iteration Retrospective

[Adjust Cookie Settings](#)

Note: For more on SAFe Scrum, please read the additional Framework articles in the Scrum series, including [SAFe Scrum](#), [Scrum Master/Team Coach](#), [Iterations](#), [Iteration Planning](#), [Iteration Goals](#), and [Iteration Review](#).

The Iteration Retrospective is a regular event where the team members discuss the results of the iteration, review their practices, and identify ways to improve.

Agile teams that apply Scrum hold a retrospective at the end of each iteration. Each retrospective seeks to uncover what's working well, what's not, and what the team can do better next iteration. Agile Teams applying Kanban hold retrospectives whenever needed, but they often use the same cadence as Scrum teams.

Details

[Agile Teams](#) reflect on the completed iteration and derive new ideas to improve the team's process. This reflection helps instill the concept of relentless improvement — a fundamental tenet of the SAFe [Core Values](#). And it helps ensure that every retrospective yields some minor improvements.

Inputs and Outputs of the Iteration Retrospective

Inputs to iteration retrospective may include:

- [Iteration goals](#)

- The team's increment

- List of improvement stories identified and the actions taken since the last retrospective

- Collection of agreed-to iteration metrics

A successful iteration retrospective event delivers the following outputs:

- Creation of a few improvement [Stories](#)

- Updated [Team Backlog](#)

Preparation

The [Scrum Master/Team Coach](#) should offer the team a way to give feedback on the techniques used at the last retrospective to improve the event's facilitation.

Process

The entire team participates in the retrospective. The Scrum Master/Team Coach starts the event by introducing the goals of the retrospective, the agenda, and the facilitation format for the meeting.

The team reviews and discusses the metrics the team has agreed to and determines any actions to take, such as further investigation or analysis.

Team members may write their thoughts on a flip chart or the digital tool designated for the retrospective. Following are several popular formats for getting started with qualitative feedback for the iteration (also see [1], [3], [4], [5]):

Individual. Individually write post-it notes and then group similar stickies

Appreciation. Note whether someone has been helpful to the team

Conceptual. Choose one word to describe the iteration

Rating. Rate the iteration on a scale of one to five, and then brainstorm how to make the next one a five

Simple. Open a discussion and record the results under the three headings described next.

The last format is the most conventional. The team creates stickies to record 1) *what went well*, 2) *what did not*, and 3) what to *do better next time* and then facilitates an open brainstorming session.

Of course, teams can examine more than three questions, match the questions to themes, and can use this time to explore new ways of getting feedback and improvement items. The team can adopt the simple format quickly, making all accomplishments and challenges visible. Optionally, the team can use different headings for the simple retrospective, as shown in Figure 1.

The last part of the process is to hold a team vote on the action items and suggestions for improvement to add to the [Team Backlog](#). If a significant issue is identified, the team may perform a root cause analysis, discuss potential corrective actions, and enter improvement stories into the team backlog. (See the *problem-solving* section in the [Inspect & Adapt](#) article for more information on root cause analysis.)

Figure 1. One team's iteration retrospective results using a simple 4Ls format [1]

Attendees

Attendees of the iteration retrospective event include:

The Product Owner (PO)

Scrum Master/Team Coach

All team members and other stakeholders or subject matter experts

Other stakeholders, which may include representatives from other Agile Teams or trains.

Scrum Masters/Team Coaches or POs typically facilitate iteration retrospectives for the team, ensuring they stay within the agreed event timebox.

Agenda

The timebox for the event is a maximum of one hour for a two-week iteration. An *example* iteration retrospective agenda (Figure 2) and a description of each item follow.

Figure 2. Example retrospective agenda

Figure 2. Example retrospective agenda

The retrospective generally has two parts:

1. **Quantitative review.** The team assesses if the iteration goals were met using a binary (yes or no) measure. Agile Teams collect and review iteration metrics for transparency and to assist with process improvement. Examples include [flow metrics](#), such as flow velocity, load and distribution, defects addressed, and automated test coverage. This data also provides the context for the qualitative section that follows.
2. **Qualitative review.** During the qualitative portion, the team reviews the improvement stories they identified in the last retrospective. They then analyze the current process, focusing on finding one or two things they can do better in the next iteration. Since many improvement items have a significant scope, the team can divide them into smaller improvement stories to be implemented incrementally in subsequent iterations.

Other Improvement Opportunities

As organizations begin implementing [DevOps](#) and a [Continuous Delivery Pipeline](#), Agile Teams will have many improvement opportunities, including:

Applying [Built-In Quality](#)

Enhancing test automation (including [test-driven development](#) and [behavior-driven development](#)) and [Continuous Integration](#)

- Automating the deployment process
- Decoupling deployment from release (see [Release on Demand](#))
- Building telemetry and recovery techniques into systems

[Lean-Agile Leaders](#) support the time teams need during each iteration to focus on cultivating new skills and addressing improvement opportunities. The [Innovation and Planning \(IP\) Iteration](#) also offers opportunities for teams to advance their skills.

Keeping Team Members Engaged

Team members are more likely to remain more engaged when retrospective formats are new and varied. For example, teams may choose to rotate the responsibility for facilitating retrospectives. One fun practice is allowing each person to select the retrospective format when it's their turn to lead. This practice creates shared ownership of the process and keeps the retrospective interesting. Having different themes and templates to focus the retrospective on specific topics can also be fun and valuable. The SAFe Collaborate tool, accessed from the community platform, provides many retrospective templates.

Learn More

[1] Sprint retrospective techniques.

<https://waynedgrant.wordpress.com/2014/02/09/sprint-retrospective-techniques-3/>

[2] Derby, Esther, and Diana Larson. *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf, 2006.

[3] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley, 2007.

[4] Fun Retrospectives. www.funretrospectives.com

[5] TastyCupcakes.org. <http://tastycupcakes.org/tag/retrospective/>

[6] Agile Retrospective Resource Wiki. <http://www.retrospectivewiki.org>

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Solutions.png 976×196 pixels

Scaled Agile Framework



Knowledge workers themselves are best placed to make decisions about how to perform their work.

ÑPeter F. Drucker

Principle #9 Ð Decentralize decision-making

[Adjust Cookie Settings](#)

Delivering value in the shortest sustainable lead time requires decentralized

decision-making, which is a critical tenet of Lean thinking. Any decision that must be escalated to higher levels of authority introduces a delay. Also, escalated decisions can decrease quality due to the lack of local context, plus changes to the facts that occur during the waiting period.

Conversely, decentralizing decision-making reduces delays, improves product development flow and throughput, and facilitates faster feedback and more innovative solutions. Higher levels of empowerment are an additional, tangible benefit.

Centralize Strategic Decisions

Of course, not all decisions should be decentralized. Some strategic decisions have far-reaching impacts and are outside the teams' scope, knowledge, or responsibilities. In addition, leaders are still accountable for outcomes. They also have the market knowledge, longer-range perspectives, and understanding of the business and financial landscape necessary to steer the enterprise.

Some decisions, then, should be centralized. Generally, they share the following characteristics:

Infrequent □ Made infrequently, these decisions typically are not urgent, and deeper consideration is appropriate (ex., product strategy, international expansion).

Long-lasting □ Once made, these decisions are unlikely to change, at least in the short term (for example, commitment to a standard technology platform, commitment to organizational realignment around Value Streams).

Provide significant economies of scale □ These choices deliver large and broad economic benefits (for example, a common way of working and standard development platforms, languages, and tooling).

Leadership is responsible for making these types of decisions, supported by the input of stakeholders affected by the results.

Decentralize Everything Else

The vast majority of decisions do not reach the threshold of strategic importance. All other decisions should be decentralized. Characteristics of these types of decisions include:

Frequent □ The problems addressed by decentralized decisions are recurrent and common (for example, Team and ART Backlog prioritization, real-time Agile Release Train [ART] scoping, response to defects and emerging issues).

Time-critical ⚠ Delaying these types of decisions comes with a high cost of delay (for example, point releases, customer emergencies, and dependencies with other teams).

Require local information ⚠ Some decisions need local context, whether technology, organization, or knowledge of a particular customer impact. Examples include: shipping a release to a specific customer, resolving a significant design problem, or self-organizing to address an emerging challenge.

Workers with local context and detailed knowledge of the technical complexities of the current situation should make these decisions.

A Lightweight Thinking Tool for Decision-Making

Understanding how decisions are made helps knowledge workers approach decision-making more confidently. Leadership's responsibility is to establish the rules for decision-making (including, for example, the Economic Framework) and then empower others to make them. Figure 1 illustrates a simple tool or exercise for deciding whether decisions should be centralized or decentralized.

Figure 1. A simple decision-making framework and exercise

Figure 1. A simple decision-making framework and exercise

Leaders can practice with this tool and understand better what types of decisions they need to make and what decisions are better left to those they lead.

Last update: 6 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe
Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



There's innovation in Linux. There are some really good technical features that I'm proud of. There are capabilities in Linux that aren't in other operating systems.

Linus Torvalds,
creator of Linux

Adjust Cookie Settings

Features and Capabilities

A Feature represents solution functionality that delivers business value, fulfills a stakeholder need, and is sized to be delivered by an Agile Release Train within a PI.

Each feature includes a benefit hypothesis and acceptance criteria and is sized or split as necessary to be delivered by a single Agile Release Train (ART) in a PI.

A Capability represents large solution functionality whose implementation often spans multiple ARTs and is sized to be delivered within a PI.

Features also lend themselves to the [Lean UX](#) process model, which includes a definition of the Minimum Marketable Feature (MMF), a benefit hypothesis, and acceptance criteria. The MMF helps limit the scope and investment, enhances agility, and provides fast feedback. Capabilities behave the same way as features. However, they are at a higher level of abstraction and support the definition and development of large [Solutions](#).

Details

Features and capabilities are critical to defining, planning, and implementing [Solution](#) value. Figure 1 provides a broader context for these work items:

Figure 1. Features and Capabilities in the SAFe context

Figure 1. Features and Capabilities in the SAFe context

Figure 1 shows that solutions are developed using features. Each reflects a service provided by the system that fulfills some important stakeholder needs. They are maintained in the [ART Backlog](#) and sized to fit in a [PI](#) so that each delivers new value. Features can originate from either the [Agile Release Train \(ART\)](#)'s local context or from splitting [Epics](#) or capabilities.

The [ART and Solution Train Kanban](#) systems support the flow of features and capabilities, where they progress through the funnel, analyzing, backlog, implementing, validating, deploying, and releasing states. This process provides reasoned economic analysis, technical impact, and strategy for incremental implementation.

[Product Management](#) and [System Architect](#) define the features and enablers, respectively. [Nonfunctional Requirements](#) (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. NFRs serve as constraints or restrictions on the system's design across the different

backlogs. Features are prioritized using [Weighted Shortest Job First \(WSJF\)](#) and are planned and reviewed at PI boundaries. They are split into [Stories](#) and are implemented, integrated, tested, and demonstrated as the functionality becomes available.

Discovering and Describing Features

[Design Thinking](#) takes a [Customer-Centric](#) approach to creating desirable and sustainable products. Design thinking tools, including personas, empathy maps, and customer journey maps, provide empathy towards and a deeper understanding of customers and users. Together, they offer a rich context to better understand features and their potential benefits.

Features are defined using a features and benefits format:

Feature ☐ A short phrase giving a name and context

Benefit hypothesis ☐ The proposed measurable benefit to the end user or business

Avoid defining features with the “[User story voice](#)” format designed to support one user role; features typically provide functionality for multiple user roles. Furthermore, using the same method to describe user stories and features may cause confusion.

Figure 2 illustrates an example set of features with benefits hypotheses:

Figure 2. Features and benefits hypotheses

Figure 2. Features and benefits hypotheses

Creating and Managing Features

In collaboration with [Product Owners](#) and other key stakeholders, Product Managers define features in the local context of an ART. Some arise as a result of splitting epics.

System Architects typically create enabler features. The ART backlog is used to maintain enablers alongside business features. Enablers pave the [Architectural Runway](#) and support exploration or provide the infrastructure needed to develop, test, and integrate the solution.

Like business features, enabler features may originate from epics or emerge locally at the ART level. Enablers that make it through the [Kanban](#) system will be subject to capacity allocation in the ART backlog to ensure enough emphasis on furthering the solution and extending the architectural runway. At each PI boundary, the percentage of resources allocated to new features (or capabilities) versus enablers

is estimated to guide the train.

Prioritizing Features

The WSJF prioritization model is used to sequence jobs (e.g., features, capabilities) based on the economics of product development flow. Since implementing the *right jobs* in the *right sequence* produces the maximum economic benefit¹ it is hard to overstate the importance of this critical process.

Product and Solution Management use WSJF to prioritize features, while System and Solution Architects also use WSJF to prioritize enabler features. Since business features and enabler features exist in the same backlog, Product and Solution Management must work collaboratively with System and Solution Architects to reconcile differences in priorities. Aligning priorities to [Strategic Themes](#) and capacity allocation are two approaches to creating alignment and balance in the backlog.

Estimating Features

Feature estimation supports forecasting value delivery, applying WSJF prioritization, and sizing epics by splitting them into features and summing their estimates. Feature estimation usually occurs in the analysis state of the ART Kanban. It relies on normalized estimation techniques, similar to the methods used by Agile teams (see the [Iteration Planning](#) article for more detail). During analysis, select subject matter experts from the ART engage in exploration activities and preliminary sizing.

Accepting Features

Feature acceptance criteria determine whether the implementation is correct and delivers the business benefits. Figure 3 provides an example:

Figure 3. Feature with acceptance criteria

Figure 3. Feature with acceptance criteria

Acceptance criteria mitigate implementation risk and enable early validation of the benefit hypothesis by creating alignment between product management, stakeholders, and developers. Acceptance criteria can also be used as the source of stories. As with stories, acceptance criteria are often transformed into acceptance tests with [Behavior-Driven Development](#) (BDD).

Product Management is responsible for accepting the features. They use acceptance criteria to determine whether the functionality is implemented correctly and whether nonfunctional requirements are met.

Capabilities

Most of this article is devoted to describing the definition and implementation of features, as they are the most common description of system behavior. Capabilities exhibit the same characteristics and practices as features. For example, they:

- Are described using a phrase and benefit hypothesis
- Are sized to fit within a PI; however, they often take multiple ARTs to implement
- Are reasoned about and approved using the Solution Train Kanban. The [Solution Train Backlog](#) holds approved capabilities
- Have associated enablers to describe and bring visibility to all the technical work necessary to support the efficient development and delivery of business capabilities
- Are accepted by [Solution Managers](#), who use the acceptance criteria to determine whether the functionality is fit for purpose

Capabilities may originate in the local context of the solution or occur as a result of

splitting portfolio epics that may cut across more than one [Value Stream](#). Another potential source of capabilities is the [Solution Context](#), where some aspects of the environment may require additional solution functionality.

Splitting Features and Capabilities

Capabilities must be decomposed into features to be implemented. They, in turn, are split into stories consumable by teams within an iteration. SAFe provides ten patterns for splitting work, as described in Leffingwell [1], chapter 6.

1. Workflow steps
2. Business rule variations
3. Major effort
4. Simple/complex
5. Variations in data
6. Data methods
7. Deferring system qualities
8. Operations
9. Use-case scenarios
10. Breaking out a spike

Figure 4 illustrates splitting a capability into features.

Figure 4. A capability split into features

Figure 4. A capability split into features

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 14 December 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile

Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



*If we knew what we
were doing, it wouldn't
be called research.*

~Albert Einstein

Spikes

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

[Adjust Cookie Settings](#)

Spikes are a type of SAFe [Enabler Story](#). Defined initially in Extreme Programming

(XP), spikes represent activities such as exploration, architecture, infrastructure, research, design, and prototyping. Their purpose is to gain the knowledge necessary to reduce the risk of a technical approach, better understand a requirement, or increase the reliability of a story estimate.

Like other stories, spikes are estimated, implemented and demonstrated. They also provide a mechanism and workflow that [Agile Release Trains \(ARTs\)](#) use to help determine the viability of [Epics](#).

Details

Agile and Lean value facts over speculation. When faced with a question, risk, or uncertainty, [Agile Teams](#) conduct small experiments before moving to implementation rather than speculate about the outcome or jump to a [Solution](#). Teams may use spikes in a variety of situations:

- Estimate new [Features and Capabilities](#) to analyze the implied behavior, providing insight into the approach for splitting them into smaller, quantifiable pieces
- Perform feasibility analysis and other activities that help determine the viability of Epics
- Conduct primary research to familiarize them with a new technology or domain
- Gain confidence in a technical or functional approach, reducing risk and uncertainty

Spikes may involve creating a small program, research activity, or test demonstrating new functionality.

Technical and Functional Spikes

Spikes primarily come in two forms: technical and functional.

Functional spikes D They are used to analyze overall solution behavior and determine the following:

- How to break down and organize work
- Where risk and complexity exist
- How to use insights to influence implementation decisions

Technical spikes D They are used to research various approaches in the solution domain. For example:

- Determine a build-versus-buy decision
- Evaluate the potential performance or load impact of a new user story

- Evaluate specific technical implementation approaches
- Develop confidence about the desired solution path

Some features and user stories may require both types of spikes. Here's an example:

As a consumer, I want to see my daily energy use in a histogram to quickly understand my past, current, and projected energy consumption.

In this case, a team might create both types of spikes:

- A **technical spike** to research how long it takes to update a customer display to current usage, determining communication requirements, bandwidth, and whether to push or pull the data
- A **functional spike** Prototype a histogram in the web portal and get some user feedback on presentation size, style, and charting

Guidelines for Spikes

Since spikes do not directly deliver user value, use them sparingly. However, they do create knowledge and learning. The following suggested guidelines apply.

Quantifiable, Demonstrable, and Acceptable

Like other stories, spikes are put in the [Team Backlog](#), estimated, and sized to fit in an iteration. Spike results may differ from a story because they often produce information rather than working code. They should develop only the information required to identify and size the stories that drive it confidently or to determine the best technical approach.

The output of a spike is demonstrable, both to the team and to any other stakeholders, which brings visibility to the research and architectural efforts and helps build collective ownership and shared responsibility for decision-making. The [Product Owner](#) accepts spikes that have been demoed and meets its acceptance criteria.

Timing of Spikes

Since they represent uncertainty in one or more potential stories, planning for both the spike and the resulting stories in the same iteration is sometimes risky. However, if it's small and straightforward, and a quick solution is likely to be found, doing both in the same iteration can be efficient.

The Exception, Not the Rule

Every user story has uncertainty and risk. That's the nature of Agile development. The team discovers the right solution through discussion, collaboration, experimentation, and negotiation. Every user story contains spike-like activities to identify the technical and functional risks. The goal of an Agile Team is to learn how to address uncertainty in each iteration. Spikes are critical when high uncertainty exists or there are many unknowns.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison Wesley, 2011.

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
 5400 Airport Blvd., Suite 300
 Boulder, CO 80301 USA

BUSINESS

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[YouTube](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



A guiding coalition that operates as an effective team can process more information more quickly. It can also speed the implementation of new approaches because powerful people are truly informed and committed to key decisions.

~John Kotter

[Adjust Cookie Settings](#)

Create a Lean-Agile Center of Excellence

This is article **three** in the **SAFe™ Implementation Roadmap series**. [Click here](#) to view the entire roadmap.

The *Lean-Agile Center of Excellence (LACE)* is a small Agile team dedicated to implementing the SAFe Lean-Agile way of working within an enterprise. An effective LACE is one of the key differentiators between companies practicing Agile in name only and those fully committed to adopting Lean-Agile practices and getting the best business outcomes. The LACE is one element of the “sufficiently powerful guiding coalition” for change, which is made up of three primary ingredients:

- [Reaching the Tipping Point](#)
- [Train Lean-Agile Change Agents](#)

This article provides guidance for the LACE’s characteristics, primary responsibilities, size, and suggested operational models based on the SAFe knowledge base and the experiences of effective LACE members in the field.

Details

The articles [Train Lean-Agile Change Agents](#), [Train Executives, Managers, and Leaders](#), and [Leading in the Digital Age](#) describe how organizations can help change agents and leadership gain the knowledge and skills necessary to lead the transformation.

The challenge is that most people qualified to enable the change have full-time responsibilities in their current roles. While a significant portion of their time may be devoted to *leading* the change, a smaller, more dedicated group of people is required to *drive* the day-to-day actions of the SAFe transformation throughout the organization. Though these groups go by different names—the Agile Center of Excellence, Agile Working Group, Lean-Agile Transformation Team, Learning and Improvement Center—they are staffed with people whose primary task is implementing and sustaining the change.

Characteristics

Effective LACES are made up of people in the organization that have some common

characteristics. When forming the first LACE, or a new LACE for a new area of the organization, it can be useful to identify those who already exemplify these characteristics:

Respect of peers ☐ Individuals in the organization who, regardless and sometimes despite their current title, have the respect of others. They already influence new behaviors and practices. They are known for getting things done.

Motivated by helping others ☐ Individuals who are happiest when they are engaged in helping others. They are motivated by seeing those around them succeed. They may not have leadership titles or management responsibilities, but they play critical roles in the organization.

Pragmatic optimism ☐ Individuals who work to find solutions to current problems and remain hopeful for the organization's future. They are not easily discouraged or shy about bringing up areas the company can improve. They are able to do so in a positive and team-oriented way.

LACE participants are talented and valuable people. Leaders must also consider any organizational and financial impact when assigning personnel to the new charter.

Organizational Model

The LACE may be a part of an organization's emerging Value Management Office (VMO), or it may exist as a stand-alone unit. In either case, it serves as a focal point of activity, a continuous source of energy that can help power the enterprise through the necessary changes. It should be placed high enough within the organization to influence new practices and behaviors. Additionally, since becoming a Lean-Agile enterprise is an ongoing journey, the LACE often evolves into a longer-term center for relentless improvement toward business agility.

As author John Kotter notes, "The size of an effective coalition seems to be related to the size of the organization." Change often starts with just two or three people. In successful transformations, this group grows to half a dozen in relatively small firms or multiple groups in larger firms. [1]. Regarding team size, experience has shown that small teams of four to six dedicated people can support a few hundred practitioners, while teams of about twice that size support proportionally larger groups. Beyond that, team size gets unwieldy, and a decentralized or hub-and-spoke model is typically more effective, as illustrated below in Figure 1.

Figure 1. LACE organizational models

Figure 1. LACE organizational models

Responsibilities

With the LACE members identified and the organizational model that best suits the current state of the transformation applied, the LACE gets to work on their critical responsibilities for supporting long-lived business agility. These responsibilities, grouped into five areas (Figure 2), enable the LACE to serve the complete SAFe transformation throughout all roles and levels of the organization.

Figure 2. Areas of responsibilities of a LACE

Figure 2. Areas of responsibilities of a LACE

Each of these five areas of responsibility is described in detail below.

Facilitating the Transformation

Organizations don't transform themselves. It requires careful, persistent attention to the most effective ways to evolve mindsets and practices within the given culture. The LACE facilitates this process by establishing a clear mission, committing to the Agile journey, building exemplary Agile capabilities, and communicating the transformation vision and current state.

Maintain a LACE mission Teams perform best when they agree on their mission. An example LACE mission statement is included in Figure 3.

Figure 3. Sample LACE mission statement

Figure 3. Sample LACE mission statement

Be an exemplar Agile Team Operationally, the LACE functions as an Agile team and applies the same iteration and PI cadences. This allows the LACE to plan, inspect and adapt in harmony with the ARTs, exemplifying Agile team behavior. As a result, similar roles are needed:

A Product Owner works with stakeholders to prioritize the team's transformation backlog. This backlog is visible and transparent to the organization.

A Scrum Master/Team Coach facilitates the process and helps remove roadblocks.

The team is cross-functional. Credible people from various functional organizations are integral members of the team. That allows them to address backlog items, whether related to organization, culture, development process or technology, wherever they arise.

A C-level leader typically acts as the team's **Business Owner**. This leader takes the lead on the early blocking and tackling needed to remove barriers to change and remains engaged enough to ensure the transformation stays aligned with the organization's strategic needs.

Illustrate SAFe expertise In addition to operating as an Agile Team, the LACE is transparent about its own SAFe journey. They participate in Implementing SAFe, becoming SPCs who understand the complete Implementation Roadmap. They use their team to work through experiments on improving flow and efficiency, creating the ability to speak from a place of experience when coaching others. They use data to inform improvements in the organization, utilizing qualitative and quantitative feedback to design future changes. The LACE will advocate Lean-Agile behaviors, focused on modeling principles and their own behaviors. They provide thought leadership and apply real-life experience to support teams, trains, and senior leadership in meeting their goals and relentlessly improving.

Communicate the transformation vision and progress The LACE, as part of the guiding coalition, uses opportunities like all-hands, syncs, company communication channels, and more to continue reporting on the transformation's progress and its continued vision. This communication is not just data transfer. Learning from Kotter, the LACE shows people something that addresses their anxieties, that accepts their anger, that is credible in a very gut-level sense, and that evokes faith in the vision.[2]

Managing the Transformation Backlog

Like any Agile team, the LACE has an important job to do and must efficiently manage the flow of work. The tasks that fulfill their mission must be clearly defined, regularly prioritized, and worked on by all members. The LACE establishes a transformation backlog to organize its work and maintains it on an ongoing basis. The backlog is transparent to the organization, is aware of the current state of agility and improves upon that baseline, and iteratively achieves transformation OKRs.

Manage a shared backlog- The LACE has a tall order to fill: change the behavior and culture of an organization. Once a LACE has formed, there will be a natural desire to accelerate progress and work through its entire backlog (Figure 4) as quickly as possible. However, trying to remove all the major organizational impediments right at the start will slow the transformation to a halt. Instead, with the support of the entire guiding coalition, the LACE empowers the organization to generate short-term wins by incrementally defining and launching ARTs. It then consolidates those gains as additional ARTs are launched. This provides the positive momentum needed to tackle the more significant organizational issues.

Figure 4. An example LACE Transformation Backlog

Figure 4. An example LACE Transformation Backlog

Create a baseline **D** The business agility assessment (see the [Measure and Grow article](#)) can help the LACE understand where a portfolio is on the road to [Business Agility](#). The LACE should baseline the assessment at the start of the transformation, continuously measure progress, and use the recommendations provided to drive the improvement backlog.

Identify transformation OKRs **D** With each PI, ART, and value stream, the gains continue to build, and the organization transforms incrementally. The LACE will partner with other SPCs, value stream managers, and the LPM to utilize OKRs. As the transformation expands, this may mean coaching on OKR writing and followthrough of outcome-based decisions. The LACE will exemplify this by identifying and using OKRs to measure the outcomes of the SAFe transformation itself. Some features within the LACE backlog will create results for some of the transformation OKRs directly; many other features will be enablers that the people and systems being transformed require to succeed.

Fostering Lean-Agile Learning

In many cases, the LACE will manage the centralized funding for continuous learning within a Portfolio or set of Portfolios. The LACE utilizes training metrics to partner with other departments such as HR and Finance, to educate the organization on the Ôway we work now.Ô SAFe SPCs will provide the SAFe training needed to kick off and maintain the transformation, as described throughout the Implementation Roadmap. The LACE, as part of the large SPC community, participates in crucial conversations, helps identify learning plans, and guides the [Continuous Learning Culture](#) through:

Inviting new thoughts **D** LACE members invite feedback from those affected by the change. They embrace conflicting opinions and strong pushback, with the philosophical stance that one who knows all the answers has not been asked all the questions. LACE members are voracious learners, connecting to peer communities worldwide to learn and grow their transformation with new findings.

Creating learning plans **D** In partnership with other SPCs, people managers, HR, and enterprise learning, the LACE creates learning plans that effectively onboard employees and suppliers into the new ways of working and mature their role-based competencies over time. In support of this, SAFe courseware is action-oriented and is utilized after class to further the change at hand. The LACE assists leaders and managers in understanding who has attended which classes and workshops and what kinds of action-based activities result.

Measuring learning **D** The LACE ties learning and training back into strategic and transformation-based OKRs. LACE members utilize quantitative and qualitative measurement to maintain feedback from learners, identify gaps in offerings that may require new learning offerings, and trend over time numbers trained related to role, ARTs, and Portfolio. As the transformation matures, the measurements from the LACE will also mature related to the change in practices based on learnings.

Coaching Leadership

The LACE is uniquely positioned to coach all forms of leaders throughout the organization. Their mission is dedicated to enabling business agility, enabling conversations around improving overall flow. To achieve this mission, they will be coaching leaders to lead by example, participating in and updating existing leadership programs, and connecting leaders to the voices of the employees and suppliers being asked to transform.

Facilitate leadership workshops **D** The LACE, as early SPCs, often facilitate Leading SAFe and Leading in the Digital Age cohorts, described in steps 3 and 4 of this roadmap series. After that, they continue to organize and facilitate leadership workshops, from educational series to vision and strategy alignment sessions. Actions often naturally emerge from these workshops to initiate 1:1 and small group coaching with leaders, opportunities embraced by the LACE when they arise. The LACE will offer guidance and coaching for OKRs and KPI usage, enabling RTEs, STEs,

Business Owners, and LPM members to communicate and amplify their value streams.

Participate in organizational leadership programs **D** Most large organizations have existing leadership programs. To enable the new skills required to lead in the digital age, LACE members partner and participate in existing organizational leadership programs, aligning transformation messaging to corporate strategies for retaining and growing top talent.

Serve the transformed **D** With a great mission comes great responsibility. To avoid becoming Òivory towerÓ agilists, the LACE utilizes their own learnings, coaching relationships, and practitioner COPs to maintain their organizational credibility.

Supporting LPM

The [Lean Portfolio Management](#) competency articulates some key responsibilities that pertain directly to the LACE. LACE personnel can help bring together the right leaders to initiate LPM adoption and facilitate getting started. LPM Agile Portfolio Operations and Lean Governance responsibilities benefit from the LACEÒs direct involvement (Figure 5). Close collaboration with others in the organization, facilitation, providing information gathered from multiple sources and partnering deeply with the VMO all advance the LPM cause.

Figure 5.

Figure 5. LACE Interactions within LPM

Facilitation of LPM events The LACE often becomes facilitation experts within their org, starting with Value Stream Identification Workshops early in the transformation. LACE members often facilitate Participatory Budgeting, Portfolio Reviews, and Portfolio Sync, gathering inputs from the VMO, RTEs, STEs, and Epic Owners. As non-decision makers, The LACE members have a unique and non-threatening way of facilitating these discussions with context.

Provide supporting data Through the activities of the other responsibilities, the LACE will have the opportunity to communicate cross-value stream needs that arise, which different roles may not see as quickly. With these insights, the LACE may serve as Epic Owners to instantiate or enhance the transformation. These may include tool or policy change recommendations or a business case for new infrastructure needs. These efforts are based on patterns seen across multiple DevOps classes and value stream mapping activities or observations from running various design thinking workshops.

Partner with the VMO The VMO, like the LACE, leverages specialized skills, knowledge, and relationships while transitioning themselves and the portfolio to a new Lean-Agile way of working. The VMO will have operational and financial data that

the LACE requires to succeed. The VMO and the LACE will often partner in the activities below or even assume responsibility when an organization has chosen not to form a VMO:

- Facilitation of portfolio events
- Develop, harvest, and apply successful ART and Team execution patterns
- Fostering decentralized PI Planning and operational excellence
- Establish objective metrics and reports progress toward business agility
- Focusing the portfolio on measuring and improving value delivery

Moving Forward

With the members of the LACE identified and the first set of Lean-Agile Change Agents trained as SPCs, the organization has the right people for the next critical move:

[Train Executives, Managers, and Leaders](#)

NEXT

Learn More

[1] John P. Kotter. *Leading Change*. Harvard Business Review Press, 1996.

[2] John P. Kotter, Dan S. Cohen. *The Heart of Change*. Harvard Business Review Press, 2002

Additional Resources

- [Implementing SAFe with SPC certification](#)
- [Leading SAFe with SA certification](#)

Last update: 9 December 2022

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite

300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Quality begins with
the intent.*

ÑW. Edwards Deming

Solution Intent

Solution Intent is the repository for storing, managing, and communicating the knowledge of current and intended solution behavior and design.

Where required, this includes both fixed and variable specifications and designs; reference to applicable standards, system models, functional and nonfunctional tests; and support for traceability.

Adjust Cookies settings

The reason for solution intent is obvious. Building large-scale software and cyber-physical systems is one of the most complex and challenging endeavors in the industry today. This requires alignment on two central questions:

What exactly is this thing we are going to build?

How will we build it?

What's more, these two questions are interrelated. If one doesn't know "how" to build something in an economically or technically feasible way, then "what" is being built must be reconsidered. SAFe labels this critical knowledge pool *Solution Intent*. It provides a basic understanding of the current and evolving requirements, design, and intent—that is, the larger purpose of the solution.

Some solution intent is fixed, with non-negotiable requirements for what it must do, or already does. Some solution intent is variable, subject to further discussion and exploration as facts surface. Understanding and navigating these differences, and allowing variability to proceed (even late into the timeline), are vital to unlocking agility in large-scale solution development.

Details

When building systems with a high potential cost of failure, the need for a more rigorous definition and validation of system behavior can be a significant barrier to Agile adoption. Although many practitioners resonate with the Agile Manifesto [1] value statement of "working software over comprehensive documentation," that concept can generate conflicting priorities for enterprises that need both.

Engineering complex and highly reliable solutions require and create large amounts of technical information. Much of it reflects the solution requirements and design—[Features and Capabilities, Stories, Nonfunctional Requirements \(NFRs\)](#), system architecture, domain-level models and designs (e.g., electrical and mechanical), system interfaces, customer specifications, tests and test results, and traceability. Other relevant information records some of the critical decisions and findings of the system. This may include information from trade studies, results of experiments, the reasoning for design choices, and other items. In many cases, this information must become part of the official record, whether by necessity or regulation.

Capture System Knowledge in Solution Intent

Solution intent is a critical knowledge repository to store, manage, and communicate "what" is being built—and, where necessary, "how" it will be built. It

serves many purposes:

- Provides a single source of truth regarding the intended and actual behavior of the solution
- Records and communicates requirements, design, and system architecture decisions
- Facilitates further exploration and analysis activities
- Aligns the *Customer*, *Agile Teams*, and *Suppliers* to a common mission and purpose
- Supports *Compliance* and contractual obligations

Capture Both Current and Future Solution Intent

Figure 1 illustrates the complex nature of solution intent:

Current and future states ☐ Developers of complex systems must constantly know two things: what exactly the current system does now and what changes are intended for a future state

Specifications, designs, and tests ☐ Knowledge of both the current and future states can be captured in any form suitableÑas long as it includes the three primary elements: specifications (documented definition of system behavior), design, and tests

Figure 1. Anatomy of Solution Intent

Figure 1. Anatomy of Solution Intent

When building systems that must behave exactly as intended—*including life-critical, mission-critical, and others governed by regulatory standards*—traceability helps confirm that the system will behave as intended. It connects the elements of solution intent to each other and to the components of the systems realizing its complete behavior. The solution intent itself is created collaboratively and evolves based on learning.

The specific elements of solution intent can be realized in many forms: from documents, spreadsheets, and whiteboard sessions to formal requirements and modeling tools, as described in [Model-Based Systems Engineering \(MBSE\)](#). But solution intent is a means to the end of building the solution, so methods for capturing it should not create unnecessary overhead and waste (see the sufficiency discussion below).

Solution Intent is Dynamic

Traditionally, a set of detailed, up-front, *fixed* requirements served to define the system. But SAFe Principle #3 – Assume variability; preserve options, tells us that defining requirements and designs too tightly up-front leads to less successful outcomes. A different approach is needed, one that supports understanding what’s known yet allows what’s unknown to emerge throughout development.

Solution intent is not a static, one-time statement: it must support the entire development process and evolve. Figure 2 contrasts a traditional early, fixed requirements decomposition with a Lean-Agile approach where requirements are detailed by Features, Stories, and ultimately a face-to-face conversation.

Figure 2. Contrasting traditional and Agile requirements approaches

Figure 2. Contrasting traditional and Agile requirements approaches

Solution intent carries with it a vision of the future system that aligns teams and their backlogs. It provides the detail needed to establish the current vision but allows teams the flexibility to explore unknowns in building the solution. The resulting knowledge provides feedback on the to-be system and the opportunity to

adapt.

Keep Options Open with Fixed and Variable Solution Intent

Solution intent serves a variety of purposes. However, none of them mandates creating fully defined “point-solution” specifications up-front. Such early decisions restrict the exploration of better economic alternatives and often lead to waste and rework [2]. To prevent this, SAFe describes two elements of solution intent, fixed and variable, that support the general adaptive requirements and design philosophy that create the best economic outcome.

Fixed intent represents the “knowns.” They may be non-negotiable, or they may have emerged during development. Examples include:

Certain performance specifications (“the pacemaker waveform shall always be as follows”)

Compliance standards (“comply with all PCI credit card requirements”)

Core capabilities defining the solution (“the autonomous delivery vehicle’s maximum cargo weight is 450 US pounds”)

Variable intent represents the elements that enable teams to explore the economic trade-offs of requirements and design alternatives that could meet a broader need. Once established, these new insights will eventually become fixed requirements and design decisions.

Moving from variable to fixed requires gaining knowledge to make decisions. Enablers are SAFe’s vehicle to explore unknowns and record knowledge and decisions in the solution intent. Following [Set-Based Design](#) practices, teams explore alternatives to arrive at an optimal economic decision. These decisions enable the development of downstream features in the [Roadmap](#) (Figure 3).

Figure 3. Moving from variable to fixed solution intent over time

Figure 3. Moving from variable to fixed solution intent over time

At each PI, teams are simultaneously building what they know while exploring what they don't yet know.

Collaboratively Develop Solution Intent

Solution intent is a collaborative effort between teams and program leadership. **Product** and **Solution Management**, along with **System** and **Solution Architects**, are responsible for the highest-level, system-wide decisions (system decomposition, interfaces, and allocations of requirements to various subsystems and capabilities). They also establish the solution intent's organizational structure to support future analysis and compliance needs. Solution intent helps drive localized decisions in the teams' backlogs, as shown in Figure 4.

Figure 4. Solution Intent evolves through collaboration

Figure 4. Solution Intent evolves through collaboration

Solution intent drives the solution roadmap and ultimately determines backlog items to execute. Solution intent begins with a vision describing the solution's purpose and critical capabilities, along with the system's nonfunctional requirements. This knowledge and the emerging roadmap and critical milestones guide teams in creating backlogs and planning their work. Both the roadmap and solution intent are filled with assumptions and must adapt to knowledge gained from execution (Figure 5). This form of collaboration replaces the traditional approach that strives to reduce uncertainty through centralized and detailed specifications and plans.

Figure 5. Developing Solution Intent

Figure 5. Developing Solution Intent

SAFe's guidance for continuous delivery validates assumptions through Minimum Viable Products (MVPs) that provide validated learning through frequent, quantifiable experiments. The validated learning in solution intent is predominately technical, but the business-focused Lean Startup principles still apply.

Connect Solution Intents across the Supply Chain

At scale, the solution intent does not stand alone. Since the system is composed of multiple subsystems, the solution intent often spans those subsystems, including information needed by suppliers. Suppliers will also often have separate and independent requirements, designs, and other specifications for their subsystem or capability. From their perspective, that is their solution intent. The ultimate (top-level) solution intent must include all relevant information across the subsystem

hierarchy to communicate decisions, facilitate exploration, align teams, and support compliance (Figure 6).

Figure 6. The requirements and design hierarchy of Solution Intent

Figure 6. The requirements and design hierarchy of Solution Intent

Create Minimal but Sufficient Documentation

Solution intent is a means to an end—*it’s a tool to guide, facilitate and communicate decisions and demonstrate compliance.* Planning the solution intent’s content, organization, and documentation strategies should begin with those ends in mind. But more is not necessarily better. The Lean-Agile community recommends keeping it “light” when documenting requirements, design, and architecture [5]. Best practices include:

Favoring models over documents → An environment of continuous change challenges a document-centric approach to organizing and managing solution intent. Models (including those produced by modern practices such as design thinking and

user-centered design) can provide easier ways to manage the information, as is further discussed in [MBSE](#).

Keeping solution intent collaborative ☰ There is Ôno monopoly on innovationÕ, and solution intent is not the exclusive domain of the Product and Solution Managers, Architects, and Engineers. Many team members participate in creating, providing feedback, and refining solution intent information.

Keeping options open ☰ Defer decisions to local concerns, and make them as late as possible. An adaptive approach to requirements and design keeps promising options open as long as it is economically feasible. Set-based design practices help to avoid committing to design and requirements too early.

Documenting items in only one place ☰ Record needed requirements and design decisions in one place, a single source of truth that serves as the repository of record for everyone and everything.

Keeping it high level ☰ Communicate at as high a level of abstraction as possible, and donÕt over-specify. Enable [Set-Based Design](#) by providing a range of acceptable values instead of fixed numbers. Describe solution behavior with intent, not specificity. Decentralize requirements and design decision-making authority.

Keeping it simple ☰ Solution intent is a method for building a product with compliance and contractual obligations. Record only whatÕs needed.

Learn More

[1] Manifesto for Software Development. [AgileManifesto.org](https://agilemanifesto.org)

[2] Ward, Allen, and Durward Sobek. *Lean Product and Process Development*. Lean Enterprise Institute, 2014.

[3] Reinertsen, Don. *Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

[4] Leffingwell, Dean, and Don Widrig. *Managing Software Requirements: A Use Case Approach*. Addison-Wesley, 2003.

[5] Ambler, Scott. *Agile Architecture: Strategies for Scaling Agile Development*. Agile Modeling, 2012. <http://agilemodeling.com/essays/agileArchitecture.htm>

Last update: 4 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Essentially, all models
are wrong, but some are
useful.*

~George E. P. Box

Domain Modeling

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

[Adjust Cookie Settings](#)

Domain Modeling is a method to describe and model entities and the

relationships between them, which collectively represent the *problem domain* space.

Derived from understanding system-level requirements, identifying domain entities and their relationships provides a basis for understanding and designing systems for maintainability, testability, and incremental development. Because there is often a gap between understanding the problem domain and interpreting requirements, domain modeling is essential to Agile development at scale. Driven partly by object-oriented design approaches, domain modeling envisions the solution as *domain objects collaborating* to fulfill system-level scenarios.

In SAFe, domain models connect to the [Team](#), [ART](#), [Solution Train](#), and [Portfolio](#) backlogs, providing a common language. The connection between domain models and [Nonfunctional Requirements](#) (NFRs) often helps identify alternative design approaches to satisfy the corresponding NFRs.

Domain modeling also enables organizations to use Agile design patterns and approaches that enhance long-term velocity. As the system design changes, updating and improving the domain model is vital to a continued understanding of the system and helps refactor code to reduce the system's complexity.

Details

Introduction

Domain modeling is an essential tool for software engineering: *if you only model one thing in Agile, model the domain*. Even a relatively small domain modeling effort is an excellent tool for reducing the complexity of system development. It helps clarify requirements and the design intent. It reflects the current understanding of *entities* and their *relationships* and *responsibilities* with the problem domain. Figure 1 shows an example of a domain model for a consumer subscription management system.

Figure 1. The domain model for a consumer subscription management system

Several different views of domain models express the essential aspects of the problem domain (see [1] in chapter 8 for more detail):

Robustness diagram ☐ a simplified UML communication and collaboration diagram. Its primary purpose is to ensure that the solution's use cases are sufficiently robust to represent its usage requirements.

CRC (Class, Responsibilities, Collaborators) cards ☐ a tool used in brainstorming sessions to help teams collaborate on product design with an object orientation.

Object role model (ORM) diagram ☐ shows the relationships between objects in a database and offers a conceptual approach to database modeling through objects and roles.

However, the most simple and common domain model is a class diagram illustrated in Figure 1. Such a diagram primarily shows the key conceptual entities and their relationships.

Effective domain modeling can only occur in the context of the system-level requirements, often captured as use cases or other means. *Nouns* from requirements become candidates for domain entities, while *verbs* may represent behaviors and their relationships. Together they form a *Common Language* (sometimes called Ubiquitous language, see [2], chapter 2) that allows engineering, business, and user representatives to speak the same language, minimizing miscommunication.

Domain Modeling in Agile at Large Scale

In large-scale Agile development, domain modeling is continuously used to support the following:

Analysis of [Epics](#)

[Backlog Refinement](#) for ARTs and Solution Trains

Design workshops at different levels

Refining [Vision](#) and [Roadmap](#) (typically in preparation for PI)

Domain modeling is generally developed and continuously refined by the [System Architect](#) in collaboration with other stakeholders to understand the impact of epics and features on the system. It's an excellent tool to prepare for PI Planning to understand the work in the upcoming PI.

The following example in Figure 2 shows how a domain model is used to clarify the impact of an epic:

Figure 2. A Domain Model Helps in understanding the scope of an epic

Requirements and domain modeling are mutually dependent. Domain modeling supports the clarification of requirements, while requirements help build and clarify the model. Moreover, the domain model should be updated once new requirements are implemented.

Figure 3. illustrates the relationship between SAFe backlog items and their impact on the domain model.

Backlog Item	Impact on Domain Model
Epic	Typically introduces new entities, relationships, and responsibilities.
Feature	Introduces new entities, typically new relationships or responsibilities.
Story	Introduces new and updated relationships, responsibilities, value objects, and service interfaces.
Enabler Epic	Typically affects implementation and design aspects of a range of entities, services, and repositories, such as underlying technology or platform, a generic life cycle of the entities, API constraints, and so on.
Enabler Feature	Introduces changes to implementation and design aspects for specific entities and services, typically for one product or system. Enabler features may also result in a shift in responsibilities or may introduce new value objects.
Refactor	It may identify individual helper or value objects from an entity and change internal interfaces between entities, protocols, or APIs.

Figure 3. Impact of SAFe backlog items on the domain model

Relationships between entities are critical to effective modeling—without them, the model is just a vocabulary of terms since they lack their collaborative context. Relationships drive practical requirements definition and design decisions (Figure 4). Relationships in a domain model are typically standard (for example, “includes,” “is a”) or concrete (for example, ML Admin “defines/patches” the mailing list in our case). When determining the relationships, it is much more important to adequately capture the connections between entities that convey the meaning of their role rather than to follow format agreements indiscriminately.

The common language resulting from domain modeling is used at all levels of the Agile organization to foster a clear shared understanding of the problem domain, requirements, and architecture—see Figure 4.

Figure 4. The common language used throughout an Agile organization

Figure 4. The common language used throughout an Agile organization

Even though it is crucial to product development, a common language has limitations that every organization should know. For example, the language of marketing materials may sometimes use terms that diverge from the standard language to emphasize certain temporal or subjective aspects associated with current market trends or challenges.

Teams that use [Behavior-Driven Development](#) (BDD) inevitably use a common language in specification workshops when defining human-readable tests.

The Domain Model and System Design Domain-Oriented vs. Alternative Approaches

Domain modeling is helpful for analysis and often a good conceptual model for system design. *Domain modeling* is one of the key design patterns/approaches that assume *deriving the solution object model directly from the problem*

domain while preserving both *behavior* and *data* (see [3]). See [2] for a systematic and detailed outline of such best practices, known by *Domain-Driven Design*. This approach provides a natural and very effective way of managing the inherent complexity of software development that is vital at scale. Figure 5, adapted from [3], chapter 2, compares the effort spent on enhancing software functionality versus complexity when the design is based on the domain, data structure, or transaction scripts.

Figure 5. A sense of the relationship between domain-oriented and data- or transaction-centric approaches.

Large-scale software solutions almost inevitably have complex domain logic. Data- or transaction-centric design approaches imply a very high maintenance cost. Nevertheless, too many organizations end up with highly intricate system designs that require much effort to enhance the system. While in some cases, such approaches may make sense—*and we will discuss those below*—most often, such a design, in reality, is based on the system architects’ and teams’ preferences rather than on business drivers.

One of the many reasons to base system design on the domain structure is to foster reasonable usage of patterns that support maintainability and enable highly

incremental, concurrent development. So, in our example of the subscription management system, domain modeling, and requirements may logically suggest that subscription methods will represent the primary source of change. Given the different scenarios for opt-in and opt-out functionality, it's logical to use a Bridge Pattern, shown in Figure 6, to isolate the area of frequent change and reduce the number of entities in the system (see [4], Appendix B).

Figure 6. A bridge pattern derived from analysis of the domain model

Figure 6. A bridge pattern derived from analysis of the domain model

This is just one example of how domain modeling can be effectively used for Commonality-Variability Analysis (CVA) to foster effective system object models. (See [5], chapter 8 for more detail on the CVA method).

Domain Modeling, System Design, and Nonfunctional Requirements

Nonfunctional Requirements, on the other hand, represent the primary reason for building system design around data structure or transaction scripts rather than the

domain model. Typically NFRs like performance or scalability may result in domain logic being spread across large SQL scripts, where too much logic is in the client-side validation scripts, and so on. Even though the use of such a transaction script approach (see [3], chapter 9) can be legitimate in some instances, it should be used as an exception rather than the rule. Few properly implemented exceptions will allow the Agile enterprise to benefit from a domain-oriented approach.

Refactoring the Model

Developing a shared understanding of the system with the help of domain modeling is an incremental process, just like creating code that implements the underlying domain logic. This means that just like the code, the domain model is also subject to refactoring as our knowledge about the system improves and new domain entities and their relations actualize. Keeping the system design and the current understanding of the problem domain updated in the domain-driven design approach is relatively simple. Refactoring of both typically happens synchronously or nearly so (see [2], part III). Designing an effective domain model is both an art and a science. Not uncommonly, great insights about the structure and relationships in the domain model emerge eventually. However, it is never late to start building the proper understanding and gradually improving the code towards it—new functionality allows it to control complexity.

Summary

Domain modeling is an excellent tool for Agile enterprises to carry out a common language and a fundamental structure for analyzing features and epics. The domain model is defined and continuously refactored as enterprise knowledge about the domain improves, and the system functionality evolves.

Domain models provide a vital link between the problem domain and the code. Domain-oriented design approaches enable controlling the growing complexity and cost of maintenance and enhancement effort. Domain modeling is a highly visual and collaborative effort that creates a shared understanding of the priorities and better ways to implement them. It typically involves [System Architects](#), [Product Management](#), [Agile Teams](#), and stakeholders. No other models cover as many aspects of Agile development at scale. *If you only model one thing, model the domain.*

Learn More

[1] Ambler, Scott. *Agile Model-Driven Development with UML 2.0*. Cambridge University Press, 2004.

[2] Evans, Eric. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley, 2003.

[3] Fowler, Martin. *Patterns of Enterprise Application Architecture*. Addison Wesley, 2002.

[4] Bain, Scott. *Emergent Design: The Evolutionary Nature of Professional Software Development*. Addison Wesley, 2008.

[5] Shalloway, Alan, and James Trott. *Design Patterns Explained: A New Perspective on Object-Oriented Design*. Addison Wesley, 2004.

Last update: 2 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

[About Certification](#)

Weekends: CLOSED

[Become a Trainer](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



Customer value is delivered at point-of-sale, not point-of-plan.

~ Jim Highsmith

Enterprise Solution Delivery

Enterprise Solution Delivery (ESD) is one of the seven core competencies for [Business Agility](#), each of which is essential to achieving it. Each competency is supported by a specific assessment, which enables the enterprise to assess its [Adjust](#) [Coordinate](#) [Influence](#) [Optimize](#) [Plan](#) [Prioritize](#) [Simplify](#) [Support](#) [Transform](#) [Value](#) proficiency. The [Measure and Grow](#) article provides these core competency

assessments and recommended improvement opportunities.

Why Enterprise Solution Delivery?

Building and evolving large enterprise solutions is a monumental effort. These systems require hundreds or thousands of engineers and are subject to significant regulatory and compliance constraints. Large software systems may host complex user journeys and experiences that cross multiple products and lines of business. Cyber-physical systems require a broad range of engineering disciplines and utilize hardware and other long lead-time items. As such, they demand sophisticated, rigorous practices for engineering, operations, and evolution.

NOTE: See the [Applying SAFe to Hardware Development](#) article for more details on applying SAFe's Lean-Agile practices to the hardware domain.

While maintaining the same levels of quality and compliance, enterprises must now deliver solutions faster than ever. Many large solution builders apply the well-known ÖVÖ model [1] that encourages large batches of specification and design activities with handoffs between them. Unfortunately, this slows development and delays feedback. When building systems with significant technical, user, and market uncertainty, delayed feedback often results in missed deadlines, cost overruns, and poor business outcomes.

Adopt SAFe's Lean-Agile Mindset, Values, and Principles

Competing and thriving in the digital age requires a different approach to building large solutions. But transitioning to a new, foreign development process can appear daunting and risky. SAFe's mindset, values, and principles provide the foundation for the new way of working and an enhanced company culture that enables faster, more reliable delivery with higher customer satisfaction and employee engagement. Together, they guide organizations on their journey toward [Business Agility](#):

A [Lean-Agile Mindset](#) ensures leaders and practitioners embrace the underlying values and principles when applying the SAFe and ESD practices. Lean Thinking shifts development from a traditional, large batch-and-queue system to continuous flow. Agile adopts an iterative development approach focused on fast feedback and learning. Collectively, the values and principles of Lean Thinking and Agile form the

DNA of everything within SAFe.

Belief in SAFe's [Core Values](#) of alignment, transparency, respect for people, and relentless improvement must be deeply held and exhibited by everyone in the enterprise. These tenets help guide the behaviors and actions of everyone participating in the new way of working and are foundational to supporting the new way of working.

Ten underlying [Lean-Agile Principles](#) inform all the roles, artifacts, and practices in SAFe and influence everyone's behavior and decision-making. Understanding their meaning and purpose is critical as organizations apply and tailor SAFe to their specific context. While not every SAFe practice applies similarly in every circumstance, the principles guide practitioners towards the goal of Lean: Øshortest sustainable lead time, with best quality and value to people and society.Ø

Five SAFe Principles particularly influence large solutions builderØs adoption of ESD practices, and they are summarized below:

[Principle #3, Assume variability; preserve options](#) addresses the inherent uncertainty in large solution development and encourages practitioners to keep options open by applying [Set-Based Design](#) and a fixed-variable [Solution Intent](#).

[Principle #4, Build incrementally with fast, integrated learning cycles](#) recognizes that new knowledge reduces uncertainty instead of highly-detailed plans and specifications. Quickly building, integrating, and, when possible, deploying new system functionality provides the technical, user, and operational feedback needed to learn and adjust.

[Principle #5, Base milestones on objective evaluation of working systems](#) encourages solution builders to evaluate progress based on objective, demonstrable evidence rather than proxies like traditional phase-gate milestones.

[Principle #9, Decentralize decision-making](#) reduces the delays inherent in escalating decisions that can dramatically slow productivity and decision quality at large scale, leading to delayed feedback and limiting innovation.

[Principle #10, Organize around value](#) reduces the functional silos that encourage large work batches, handoffs, and delays and creates optimal [Agile Teams](#) and [Development Value Streams](#) that can deliver value faster, more predictably, and with higher quality.

Ten Practices for Enterprise Solution Delivery

The ESD competency describes ten best practices for applying Lean-Agile development to build and advance some of the worldØs most important solutions. The three dimensions in Figure 1 group these ten practices.

Figure 1. Enterprise Solution Delivery

Figure 1. Enterprise Solution Delivery

Lean Systems Engineering applies Lean-Agile practices to align and coordinate all the activities necessary to specify, architect, design, implement, test, deploy, evolve, and ultimately decommission these systems. These practices are:

- Specify the solution incrementally
- Apply multiple planning horizons
- Design for change
- Frequently integrate the end-to-end solution
- Continually address compliance concerns

Coordinating Trains and Suppliers manages and aligns the extended and often complex set of value streams to a shared business and technology mission. These practices are:

- Use Solution Trains to build large solutions
- Manage the supply chain

Continually Evolve Live Systems ensures large solutions and their development pipeline support continuous delivery. These practices are:

- Build an end-to-end Continuous Delivery Pipeline (CDP)
- Evolve deployed systems
- Actively manage artificial intelligence/machine learning systems

The remaining sections describe these ten practices.

Specify the Solution Incrementally

Large solutions builders use specifications to manage requirements and designs, communicate them to system builders, and support compliance. The common specification process follows the ÔVÕ model (left side of Figure 2), which typically results in large batches of upfront specification work that delay implementation and feedback. While all the ÔVÕ model activities (e.g., specifying requirements) are still critically important to solution development, in SAFe, engineers perform them concurrently and in smaller batches.

To support frequent change, Solution Managers and Architects use the [Solution Intent](#), [Backlogs](#), and [Roadmaps](#). Solution intent communicates the solutionÕs current requirements and design decisions. Some are fixed and known early (see 100 Shall statements Figure 2 below). Others may vary (see many questions) and become fixed over time as teams explore alternatives ([Set-Based Design](#)) to find optimal implementations. Backlogs and roadmaps manage the work that helps reduce uncertainty and build the solution as assumptions are validated.

Figure 2. Specify the system continuously and in smaller batches

Figure 2. Specify the system continuously and in smaller batches

By managing and communicating a more flexible approach to the system's current and intended structure and behavior, the Solution Intent aligns all solution builders to a shared direction. Its companion, [Solution Context](#), defines the system's deployment, environmental, and operational constraints. Together, they align teams and provide the necessary information for compliance. Teams use the Solution Intent to drive their backlogs and localized decision-making, as shown in Figure 3. In Agile development, backlogs and face-to-face conversations replace part of the traditional, detailed requirements hierarchies.

Figure 3. Replace detailed requirements specifications with backlogs and roadmaps

At scale, the Solution Intent and Context are connected across the supply chain (Figure 4). While some decisions are fixed early, others vary

As downstream subsystem and component teams implement decisions, the knowledge gained from their [Continuous Exploration \(CE\)](#), [Integration \(CI\)](#), and (where possible) [Deployment \(CD\)](#) efforts provide insights back upstream, moving decisions from variable to fixed.

Figure 4. Incremental feedback up and down the supply chain evolves the solution intent and roadmaps

Apply Multiple Planning Horizons

Fixed, detailed plans are difficult to adjust because they discourage the quick response to change required when building innovative solutions. Instead, Agile practitioners replace detailed schedules with [Roadmaps](#) to manage and forecast work and quickly adjust them as new facts emerge. Multiple planning horizons provide the proper separation between those responsible for setting the solution's longer-term vision and milestones while allowing the ARTs and teams who build the solution to detail and plan their work.

Figure 5 shows how multiple horizons of planning apply to [Solution Trains](#). A multi-year solution roadmap of milestones and epics informs the PI Roadmaps that decompose nearer-term epics into [Features and Capabilities](#). Multiple planning horizons allow the people closest to the work to create their own plans, guided by the larger plan for the overall solution.

Figure 5. Multiple planning horizons facilitate realistic planning

Figure 5. Multiple planning horizons facilitate realistic planning

Design for Change

Architectural decisions are critical economic choices because they significantly impact the effort and cost required for future changes. Loosely coupled architectures are the strongest predictor of continuous delivery [2]. A good architectural design enables ARTs and teams to independently develop and release “Value Streamlets” (solution components within a larger value stream), as shown in Figure 6. The components are described below:

1. The Routing and Scheduling components are allocated to cloud services and support continuous updates
2. The Mobile Delivery Application is also software-based and supports continuous delivery
3. The Vehicle Control and Navigation components run on programmable hardware (CPUs and FPGAs) in the vehicle and can be updated over-the-air

4. Hardware sensors are mounted on the chassis and are rarely updated because they require taking the vehicle out of service and possible recertification

Figure 6. Architecture impacts independent and frequent value delivery

Figure 6. Architecture impacts independent and frequent value delivery

Solution designs must balance the needs of operations and development. Design choices often optimize for operations to reduce unit, manufacturing, and delivery costs without fully understanding the total costs of delayed value delivery. For example, due to their lower unit costs and power consumption, many automotive control systems use custom, application-specific integrated circuits (ASIC) electronic parts instead of programmable ones (CPUs, FPGAs). However, the logic is fixed during ASIC manufacturing and cannot be reprogrammed, significantly increasing the cost of changes. Optimal designs balance both concerns, as shown in Figure 7.

Figure 7. Solution designs should balance operational and development costs

Figure 7. Solution designs should balance operational and development costs

Frequently Integrate the End-to-End System

Specifying and building the solution in smaller batches enables developers to integrate the end-to-end system more frequently. Frequent integration provides faster learning on the technical, user, and market assumptions and risks inherent in large solution development.

Many practices support frequent integration. Applying a common cadence across all system builders and using [Solution Demos](#) as “pull events” [3] ensures the entire system is learning, not only the individual components (see [SAFe Principle #6, “Build incrementally with fast, integrated learning cycles”](#)). [Built-in Quality](#) practices foster frequent integration for all component types, including software, hardware, IT systems, and cyber-physical systems. A [System Team](#) often provides the unique knowledge and skills necessary to integrate the end-to-end solution and reduce the cognitive load on the ARTs and other teams. (Figure 8).

Figure 8. Frequent integration provides fast feedback

Figure 8. Frequent integration provides fast feedback

When full integration is impractical, partial integration lessens risks. Through partial integration (Figure 9), Agile teams and trains integrate and test in a smaller context using virtual/emulated environments, mockups, simulations, and other models described below. Early integration often leverages proxies. For example, significant IT components can be proxied with test doubles, and cyber-physical components can leverage prototypes, development kits, and breadboards.

Figure 9. Integrate the end-to-end solution at least every PI

Manufacturing costs and other constraints of long-lead-time items impact the economics of frequently creating and integrating new parts for cyber-physical systems. Fortunately, virtual modeling and rapid prototyping innovations provide environments for faster learning. Modeling and simulation innovations for Digital Shadows and Digital Twins (see [MBSE](#)) enable early learning and decision-making. And in situations requiring physical parts for feedback, rapid manufacturing service providers(PCB-In-A-Day) can quickly produce electrical, mechanical, and other physical components (Figure 10). See [4] to hear how one company's approach is reducing cycle time and cost by a factor of 10 to accelerate learning.

Figure 10. Shift learning left to accelerate design and implementation

Continually Address Compliance Concerns

Typically, any large solution failure has unacceptable social or economic costs. To protect public safety, avoid lawsuits and prevent unwanted press coverage, these

systems must undergo routine regulatory oversight and satisfy various compliance requirements.

Organizations rely on quality management systems (QMS) to dictate practices and procedures and confirm safety and efficacy. However, most QMS systems were created **before** Lean-Agile development and, therefore, were based on the conventional approaches that often assumed (or even mandated) early commitment to:

- Unvalidated specifications and design decisions
- Detailed work breakdown structures
- Document-centric, phase-gate milestones

In contrast, a Lean QMS makes compliance activities part of the regular flow of value delivery (Figure 11). For more information on Agile compliance, see the SAFe Extended Guidance article [Achieving Regulatory and Industry Standard Compliance with SAFe](#).

Figure 11. Move to a Lean QMS

Figure 11. Move to a Lean QMS

Use Solution Trains to Build Large Solutions

Ultimately, it's people who build systems. SAFe ARTs and Solution Trains define proven structures, patterns, and practices that coordinate and align the many developers and engineers who define, build, validate, and deploy large solutions. ARTs are optimized to align and coordinate significant groups of individuals (50-125 people) into a team of Agile teams—Solution Trains that scale ARTs to build very large solutions with hundreds of developers and suppliers (Figure 12).

Figure 12. Solution trains align ART and Suppliers

Figure 12. Solution trains align ART and Suppliers

As solutions scale, alignment becomes critical. Delays in ARTs and teams accumulate, leading to missed deadlines and sizable cost overruns. Solution Trains align ARTs through [Coordinate and Deliver](#) practices. To validate customer, business, and technical assumptions, They integrate the end-to-end solution at least every PI to validate customer, business, and technical assumptions. Teams and ARTs building components with longer lead times (e.g., packaged applications, hardware)

still deliver incrementally through proxies (stubs, mockups, prototypes, and models), integrating with the overall solution and supporting early validation and learning.

Manage the Supply Chain

Most large solutions development efforts include internal and external [Suppliers](#) who bring unique expertise and existing solutions to accelerate delivery. Critical to solution success, these strategic partners should operate like an ART by participating in SAFe events (planning, demos, I&A), using backlogs and roadmaps, and adjusting to changes. [Agile Contracts](#) encourage collaboration.

To operate like an ART, suppliers must utilize ART-like roles. The supplier's Product Manager and Architect continuously align backlogs, roadmaps, and architectural runways with the overall solution. Similarly, System Teams from both organizations share scripts and context to minimize errors and delays with integration handoffs.

Supply chains can become complicated. Figure 13 shows an automotive supply chain where a solution in one context is part of a large solution in another. To ensure alignment, all Product Managers must continually synchronize their roadmaps to maximize each PI's value, adjusting together as new facts emerge.

Figure 13. Complex supply chains are system-of-systems

Figure 13. Complex supply chains are system-of-systems

There are known patterns for coordinating suppliers who deliver to multiple customers. Figure 13 shows three approaches for the Autonomous driving solution from Figure 14 to support its three customers. ÔClone-and-own,Õ a common technique, creates a custom solution for each customer. Although copying prior solutions can speed delivery, it prevents economies of scale, often raises costs, and lowers quality.

A ÔplatformÕ approach treats the solution like a product, with a vision and roadmap. To create a single solution (or product line of solution variants) for multiple customers, all solution builders work from one value stream. The Ôinternal open-source modelÕ blends the two by embedding Enabling Teams (see Topologies in Agile Teams) from the platform inside the customer value stream or approving individuals to make changes and commit them back into the platform. This way, dependent value streams can meet their delivery needs while maintaining the benefits of a platform.

Figure 14. Patterns for coordinating dependent supply chains

Figure 14. Patterns for coordinating dependent supply chains

Build an End-to-End Continuous Delivery Pipeline

Continuous integration is the heartbeat of continuous delivery — the forcing function that verifies changes and validates assumptions across the entire system. Agile software teams invest in automation and infrastructure that frequently builds, integrates, and tests small developer changes, as shown in the software portion of Figure 15 below.

Large solutions are far more challenging to integrate continuously because:

- Long lead-time items may not be available
- Integration spans multiple system organizational boundaries
- Automation is rarely feasible end-to-end
- The laws of physics dictate limits for integrating physical system elements

Still, the Agile goal is to verify and validate changes quickly for fast feedback. Different components will utilize different [Continuous Delivery Pipelines \(CDP\)](#), as shown in Figure 15. To support faster learning for cyber-physical components, a hardware CDP uses the digital engineering and rapid prototyping techniques discussed earlier in the frequent integration section.

Figure 15. Different types of components utilize different CDPs

Figure 15. Different types of components utilize different CDPs

Many technologies enable the pipelines. While the software technologies are well-known and becoming standardized, the hardware community is just beginning to leverage emerging hardware technologies (Figure 16).

Figure 16. Software and hardware technologies enable the CDP

Figure 16. Software and hardware technologies enable the CDP

Evolve Deployed Systems

Because large solutions deliver significant value for many decades, they require continuous investment to support changing technology and business needs. A typical project-based approach to development invests in the initial system development but requires separate “modernization efforts” to update them. A product-based approach to development recognizes solutions evolve continuously and funds a [Development Value Stream](#) that continually flows value to customers.

Leaders define a product vision and roadmap to accelerate time to market and simultaneously build the solution and the CDP infrastructure necessary to evolve it (Figure 17). Over their lifecycle, products follow the typical S-curve adoption (see [Solution](#)), with innovation and [Design Thinking](#) occurring throughout to evolve the solution. A fast, economical delivery pipeline allows organizations to release a minimum viable solution early and advance it later.

Figure 17. Build the solution and the CDP together

Figure 17. Build the solution and the CDP together

Actively Manage Artificial Intelligence/Machine Learning systems

Practices in [AI](#), [machine learning \(ML\)](#), and [data science](#) are rapidly maturing and being used onboard solutions to control behavioral logic and offline to improve performance and delivery. The entire enterprise also uses the data generated by these large solutions to gain business insights. Solution builders have four critical considerations when applying AI and ML to these solutions:

Telemetry □ Build telemetry into the solution and the operational infrastructure to collect critical information about the system, its users, and the operating environment. Large solutions can provide data vital to improving products, optimizing operations, and understanding customers and markets better (see [Big Data](#) for more information).

Data Management Ⓜ Large solutions generate massive volumes of data at high velocities. Solution Architects collaborate with data scientists to determine what data to store, where to store it (onboard or offline), and for what duration.

Design for AI and ML Ⓜ AI/ML algorithms are replacing humans and fixed code logic to drive solution decision-making and behavior. However, these algorithms require massive datasets and significant computing power. Solution designs for AI/ML must, therefore, balance where to put the power and algorithm logic:

Onboard the solution close to the data but with expensive computing resources
or

Offline with limited data and slower response times but access to massive, cheap computing resources

Tune models Ⓜ Unlike traditional code with fixed logic, AI/ML algorithms change their behavior over time. Algorithms change themselves with data sets that grow during operations. These applications, and the models they are built on, require manual supervision to ensure they do not drift from intent. Data scientists must monitor algorithms for their predictive quality and retrain models when they drop below a certain threshold.

Accelerating Flow of Large Solution Delivery

As explained by [Principle #6 Ⓜ Make value flow without interruptions](#), SAFe is a flow-based system that promotes fast, efficient value delivery. Flow occurs when Agile teams, ARTs, and the portfolio can deliver high-quality products and services with minimal delay.

Improving flow is critical in large solution development to reduce waste and accelerate delivery for these significant investments. Bottlenecks, impediments, and redundant activities are continuously detected and remediated at all levels of SAFe to accelerate flow throughout the value stream. The ten ESD practices identify common inefficiencies that the flow accelerators described in the [Solution Train Flow](#) article can help identify and correct.

Summary

Agile development has shown the benefits of delivering early and often to generate frequent feedback and develop solutions that delight customers. Organizations must apply the same approach to larger and more complicated systems to stay competitive. ESD builds on the advances in Lean systems engineering and emerging technologies to provide a more flexible approach to the development, deployment, and operation of large solutions.

ESD also describes the necessary adaptations to create a CDP in a cyber-physical environment by leveraging simulation and virtualization. This competency also provides strategies for maintaining and updating these true living systems to extend their life and continually deliver higher value to end-users.

Learn More

[1] Systems Engineering Body of Knowledge (SEBoK).

https://www.sebokwiki.org/wiki/System_Life_Cycle_Process_Models:_Vee

[2] 2017 State of DevOps Report. <https://puppet.com/resources/report/2017-state-devops-report/>

[3] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. Amacom, 2010.

[4] The Genius of 3D Printed Rockets. <https://www.youtube.com/watch?v=kz165f1g8-E&t=479s>

[5] Ward, Allen, and Sobek Durward. *Lean Product and Process Development*. Lean Enterprise Institute, 2014.

[6] Reinertsen, Don. *Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

Last Update: 12 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe
Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT &
TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE,
INC

CONTACT US

5400 Airport Blvd., Suite
300
Boulder, CO 80301 USA

BUSINESS
HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



Generate alternative system-level designs and subsystem concepts. Rather than try to pick an early winner, aggressively eliminate alternatives. The designs that survive are your most robust alternatives.

—Allen C. Ward

Adjust Cookie Settings

Principle #3 Ð Assume variability; preserve options

Solution development is an inherently uncertain process. Technical variability and market variability are present throughout the development process. By definition, innovative new systems have never been developed, so there is no guaranteed path to success. If there were, it wouldnÕt be innovation. ThatÕs why we love this business.

System developers are naturally inclined to reduce variability as quickly as possible. The more deterministic things are, the better we feel. ThatÕs just human nature. It seems that the more we think we know and have already decided, the further along we think we are. And thatÕs true up to a point, but variability is still present.

Variability is inherently neither bad nor goodÑit just is what it is. It is the economics associated with the timing and type of variability that determines outcomes. The goal is to *manage variability*, and *preserve options*, providing the controls and flexibility teams need to build great solutions.

This article describes managing variability and preserving options with set-based design. For more on managing variability, see [Principle #7ÑApply cadence; synchronize with cross-domain planning](#).

Development Occurs in an Uncertain World

Acknowledging the continued presence of variability in development causes us to reexamine our approach. Traditional, sequential, stage-gated development practices drive developers to quickly converge on a Ôpoint

-basedÕ single optionÑan agreed to point in the requirements and design solution spaceÑand then modify the design until it meets the system intent. Everyone feels good that they have the right requirements and the right design. At least for a while.

However, the probability that it is the right starting point is low. Subsequent development efforts to make that solution work waste time and lead to significant delivery problems, as illustrated in Figure 1 [2].

Figure 1. Picking a point solution too early in the cone of uncertainty

Figure 1. Picking a point solution too early in the cone of uncertainty

There just isn't enough time to recover. The reason is that we are operating early in the "cone of uncertainty" [3] and attempting to force certainty by freezing requirements and design. The bigger and more technically innovative the system is, the higher the odds are that the agreed starting point was not the best one. And unfortunately, as that risk becomes more and more apparent, we often turn to even tighter specifications, even earlier in the discovery process. The problem isn't resolved; it's compounded.

Preserve Options with Set-Based Design

A better approach, referred to as Set-Based Design (SBD) or Set-Based Concurrent Engineering (SBCE), is illustrated in Figure 2 [4].

Figure 2. Set-Based Design provides multiple design options

Figure 2. Set-Based Design provides multiple design options

In this approach, developers initially cast a wider design net, considering multiple design choices at the start. After that, they continuously evaluate economic and technical trade-offs—typically exhibited by the objective evidence presented at integration-based learning points. Then, as Figure 3 illustrates, they eliminate the weaker options over time and ultimately converge on a final design based on the knowledge gained to that point.

Figure 3. Set-based designÑcoupled with cadence-based learning milestonesÑ produces better outcomes

This process leaves the design options open for as long as possible, converges when necessary, and produces more optimal technical and economic outcomes.

(Note: Due to big systemsÑ scope and economic impact, set-based design is a fundamental construct of [Large Solution SAFe](#). For more information, including guidance on applying set-based design to fixed schedule commitments, planning impact, and economic trade-offs, read the [Set-Based Design](#) article and the [Enterprise Solution Delivery](#) competency articles.)

Learn More

[1] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas, 2009.

[2] Lansiti, Marco. *Shooting the Rapids: Managing Product Development in Turbulent Environments*. California Management Review, 38. 1995.

[3] McConnell, Steve. *Software Project Survival Guide*. Microsoft Press, 1997.

[4] Ward, Allan C., and Durward Sobek. *Lean Product and Process Development*. Lean Enterprise Institute Inc., 2014.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*Assume variability;
preserve options.*

NSAFe Lean-Agile
Principle #3

Set-Based Design

Set-Based Design (SBD) is a Lean development practice that keeps requirements and design options flexible for as long as possible during the development process.

Instead of choosing a single-point solution upfront, teams use SBD to explore multiple options simultaneously, eliminating poorer choices over time. This

approach enhances flexibility and design outcomes by committing to technical solutions only after validating assumptions.

Details

System development involves continuously converting uncertainty into knowledge. With SBD, teams make design decisions only after they have gained sufficient knowledge and data. They maintain multiple requirements and design options open for a longer time in the development cycle. As the timeline advances, they use experiential data to narrow the focus on the final design option. Through this approach, they embrace [Principle #3 Ð Assume variability; preserve options](#) for as long as possible, providing maximum flexibility.

Set-based design is different than a traditional Òpoint-basedÓ approach. A point-based design approach commits to a set of requirements and a single design strategy too early in the Òcone of uncertaintyÓ (Figure 1). It often leads to incorrect assumptions and late discoveries that require significant rework. As the facts become known and the deadline approaches, teams must often rush to solve problems. This emergency work can result in shortcuts, stress, quality compromises, and missed commitments and deadlines.



Figure 1. A point-based design forces decisions too early in the cone of uncertainty

In a Set-Based Design approach (Figure 2), developers initially consider multiple design choices. Teams begin researching, developing, and evaluating various designs, exploring technical and economic trade-offs. Based on their work, objective evidence is available at integration-based learning points. Then, as Figure 2 illustrates, teams eliminate the weaker options over time and ultimately converge on a final design.

Figure 2. Set-based design converges towards the best solution as uncertainty decreases

As described further in references [1] and [2], SBD is essential to achieve economic efficiency in Lean product development. Figure 3 summarizes the conceptual difference between set- and point-based design approaches.

Figure 3. Comparing point-based and set-based design approaches

Figure 3. Comparing point-based and set-based design approaches

Set-Based Design Increases Economic Efficiency

Teams use SBD along with a hypothesis-driven Minimum Viable Product (MVP) approach and gain fast feedback on their choices through their [Continuous Delivery Pipeline](#). Design alternatives include the hypothesis and a set of assumptions. Teams also define experiments to gain the knowledge needed to validate or invalidate those hypotheses, eliminate poor choices, and arrive at the optimal design.

Figure 4 illustrates an example in which an autonomous vehicle's designers must select a technology to prevent forward collisions before a significant milestone.

Figure 4. Exploring alternatives for a new *forward collision prevention* vehicle subsystem

With their corresponding hypothesis statement, teams create **Enablers** that explore cost trade-offs—*for example, providing support for environmental and weather conditions, vehicle design impact on manufacturing, obstacle detection quality, Nonfunctional Requirements (NFRs), and more.* Based on validated learning, teams select the best design and record the results in the **Solution Intent** repository.

Of course, exploring multiple design choices comes at a cost, even if the designs are primarily **model-based** or paper-prototypes. (*Note: Reinertsen points out that maintaining various design options is a type of u-curve optimization, and sometimes the optimum number of choices is one.* [3])

However, a set-based design is more cost-efficient overall if there's a high degree of technological innovation, variability in market or customer conditions, or fixed deadlines. In this case, SBD design efficiency depends on several factors:

Flexibility □ Preserving a broad set of design options for as long as possible

Cost ☰ Minimizing the cost of multiple options through modeling, simulation, and prototyping

Speed ☰ Facilitating fast earning through early and frequent validation of design alternatives

The following sections provide recommended practices for achieving efficiency with SBD.

Increase Flexibility in Interfaces and Design

Teams build and integrate complex systems from subsystems and components to produce the desired system behavior. To support SBD, subsystem interfaces must be flexible at the intersection points. System engineers may specify ranges for requirements and designs for these intersection points, which they can discuss with [Product Management](#) as more validated learning occurs. For example, how would the overall system value improve if a system engineer allocated more space, weight, and power to one component instead of another?

This more experimental approach allows system designers to manage the system-level allocations, creating a collaborative environment for system-level learning, negotiation, and making sound economic choices.

Leverage Modeling, Simulation, and Prototyping

Modeling, simulation, and prototyping provide the initial learning points that help eliminate some design alternatives and confirm others. As described in the [MBSE](#) article, modeling uses a broad set of techniques specific, including digital twins, CAD, design thinking, and user-experience design options. Teams should use various SBD approaches to the parts of the system where the risk is highest, optimizing the overall cost of maintaining design alternatives.

Collapse Design Options with Frequent Integration Points

When teams explore new designs during development, uncertainty abounds, and validated learning is scarce. The best way to resolve this variability is to test design alternatives through early and frequent system integration. In part, integration points are driven by regular [System Demos](#), which occur each iteration, and [Solution Demos](#), which happen on the [PI](#) cadence. (See “[Frequently Integrate the System](#)”

in [Enterprise Solution Delivery](#).)

These frequent integration points support experiential learning, providing new insights and reducing options as the system evolves (Figure 5).

Figure 5. Frequent integration provides critical learning points that narrow design alternatives

Take a Systems View

Significant design decisions often span parallel development initiatives and require some consideration for the future and the present. For example, the ODS technology decision for the autonomous vehicle should consider more than the current initiative to avoid front collisions. Figure 6 illustrates how architects guide the teams' understanding of the larger context when making significant technology decisions.

Figure 6. Technology decisions must look beyond the current initiative

Figure 6. Technology decisions must look beyond the current initiative

As mentioned above, set-based design has a cost. Architects and teams must balance the possibility of over-engineering the solution with the need to be prepared for near-term capabilities.

Make Decisions Based on Economic Trade-offs

Design choices have different financial implications. Effectively applying SBD requires knowledge of the broader goals of the system. As described in [Principle #1](#) [D Take an economic view](#), and Reintersen [3], making decisions based on economics requires an understanding of the trade-offs (Figure 7) between a set of related factors:

Figure 7. Economic trade-offs help quantify design options

Figure 7. Economic trade-offs help quantify design options

Development expense ▷ the cost of labor and materials required to implement a capability

Lead time ▷ the time needed to implement the capability

Product cost ▷ the manufacturing cost (cost of goods sold) and deployment and operational costs

Value ▷ the economic worth of the capability to the business and the customer

Risk ▷ the uncertainty of the solution's technical or business success

Trade-offs between these factors help illustrate which design options provide the most benefits. For instance, in the earlier collision prevention example, understanding the balance between the accuracy of various detection technologies vs. their added manufacturing cost can dramatically change the decision, as shown in Figure 8.

Figure 8. A trade-off between cost and performance (error margins in this case) helps select the best designs alternatives

Learn More

- [1] Ward, Allen, and Durward Sobek. *Lean Process and Product Development*, 2nd edition. Lean Enterprise Institute, Inc., 2014.
- [2] Oosterwal, Dantar P. *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. Amacom, 2010.
- [3] Reinertsen, Don. *Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

Last update: 31 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



In order for you to keep up with customer demand, you need to create a deployment pipeline. You need to get everything in version control. You need to automate the entire environment creation process. You need a deployment pipeline where you can create test and production

Adjust Cookie Settings

*environments, and
then deploy code
into them, entirely
on demand.*

ÑErik to
Grasshopper, *The
Phoenix Project*

Continuous Deployment

Continuous Deployment (CD) is an aspect of the Continuous Delivery Pipeline that automates the migration of new functionality from a staging environment to production, where it is made available for release.

CD is the third aspect in the four-part [Continuous Delivery Pipeline](#) (CDP) of [Continuous Exploration](#) (CE), [Continuous Integration](#) (CI), Continuous Deployment (CD), and [Release on Demand](#) (Figure 1).

Figure 1. Continuous deployment in the context of the CDP

Figure 1. Continuous deployment in the context of the CDP

Features must be available and verified in production **before** the business needs them to support [Release on Demand](#). Therefore, it's optimal to separate the deployment process from releasing, enabling changes to move into production without affecting the behavior of the current system. Continuous deployment allows teams to deploy small, incremental changes to production continually.

The capability to continuously deploy is critical for releasing on demand. In turn, it allows Agile Release Train (ARTs) to respond to market opportunities with the highest possible value in the shortest sustainable lead time, permitting customers to consume new functionality when they are ready.

Details

Traditional development practices treat deployment and release as the same activity. In this model, changes deployed to production are immediately available to users. Continuous deployment, however, separates the deployment and release

processes. This practice fosters design thinking and fast value flow by:

Targeting functionality to specific customers **D** Enables the organization to target customers with particular functionality, allowing the organization to assess the impact of changes before deploying functionality to all customers.

Promoting experimentation, such as A/B Testing **D** Design thinking practices, such as A/B testing, require the ability to present different functionality to distinct target users, gathering feedback that helps create the optimal user experience.

Promoting small batches **D** Automating the CDP (for example, tests, builds, deploys) makes deploying in small batches economically feasible.

Releasing on business needs **D** ARTs tend to release less frequently when the deployment process is complex and error-prone. Organizations that invest in automation and relentless process improvement can release faster and with lower risk, substantially increasing [Business Agility](#). For example, a release can be deployed in production ahead of a marketing campaign, giving the organization more flexibility in maximizing all aspects of value delivery.

To enable these capabilities, ARTs focus on reducing the transaction cost and risk of moving changes to production by automating all aspects of continuous deployment. Ensuring the deployment process is a repeatable, predictable activity without significant incidents helps teams achieve continuous deployment. Moreover, improving deployment to make value flow without interruption [Principle #6](#)) is critical for achieving business agility. See the [ART Flow](#) article for more information.

The Four Activities of Continuous Deployment

SAFe describes four activities of Continuous Deployment, as illustrated in Figure 2.

1. **Deploy** **D** the practices necessary to deploy a solution to a production environment
2. **Verify** **D** the practices needed to ensure solution changes operate in production as intended before releasing them to customers
3. **Monitor** **D** the practices to monitor and report on any issues that may arise in production
4. **Respond** **D** the practices to address any problems rapidly which may occur during deployment

Figure 2. Four activities of continuous deployment

Figure 2. Four activities of continuous deployment

Deploy

Deployment is the migration of changes into a production environment. In the CDP, deploying changes is done continuously. Partial functionality can be implemented incrementally into production.

Suppose a user story map has 27 stories to implement a new workflow. Traditional legacy practices would likely result in all 27 stories deployed in one large batch. Instead, individual stories can be deployed in “dark mode” with continuous deployment and feature toggles. When the ART has deployed the full features set, the business can choose when to release them to users.

Ideally, the deployment pipeline triggers the deployment process automatically following a successful build, integration, and validation. This approach makes the workflow a fully-automated one-click process, from code-commit to production-deploy. Highly sophisticated enterprises can reliably deploy anytime, even during peak periods. This approach eliminates the need to work weekends, nights, or

other off-hours to deploy.

Several practices contribute to the ability to deploy:

Dark launches ▷ the ability to deploy new functionality to a production environment without releasing it to end users

Feature toggles ▷ a technique to facilitate dark launches by implementing toggles in the code, which enables switching between old and new functionality

Deployment automation ▷ the ability to deploy a tested solution automatically from check-in to production

Selective deployment ▷ the ability to deploy to specific production environments and not others based on criteria such as geography, market segment, and more

Self-service deployment ▷ when automated deployment is partially implemented, self-service allows a single command to move solutions from staging to production

Version control ▷ maintaining environments under version control enables fast deployment and recovery

Blue/green deployment ▷ a technique that permits on-demand switching between staging and production environments

Verify

Deployments must be verified for functional integrity and robustness before releasing to end users. These two processes almost happen simultaneously when tightly coupled, making recovery decisions a primary concern. However, when they are separated, there's room to test new functionality extensively in production *before* approving it for release. After migration to production, solutions undergo a final round of testing. Typically, this requires a smoke test, light user acceptance testing, and a stress and performance test, which must occur in a production setting. This verification provides the necessary sanity check that tests the behavior of the solution in its actual production [Solution Context](#).

[Continuous Integration](#) reasonably ensures that the solution will behave as expected in production; however, surprises do occur. When verification reveals critical defects, deployments must either be rolled back or fixed quickly to prevent them from harming the production environment or disrupting the business flow.

Four practices help drive verification after deployment:

Production testing ▷ testing solutions in production using feature toggles or dark launches

Test automation ▷ the ability to automate tests and run them rapidly and repeatedly

Test data management ▷ managing test data in version control to ensure consistency in automated testing

Testing nonfunctional requirements (NFRs) ▷ teams also test system attributes

such as security, reliability, performance, scalability, and usability to ensure NFRs meet quality standards

Monitor

Verifying that deployed features didn't break on their way into production is an essential pre-release quality check. However, teams must also ensure they can measure a feature's performance and value over its lifespan. The insights that drive this critical feedback loop primarily come from robust monitoring capabilities, which must be in place before release.

Effective monitoring requires that full-stack telemetry is active for all features deployed through the CDP. This telemetry allows teams to verify system performance, end-user behavior, incidents, and business value rapidly and accurately in production. The data collected provides tracking and monitoring of each feature, increasing the fidelity of analysis of the business value delivered and increasing responsiveness to production issues.

While teams cannot collect some business-value metrics until release, they need to know how to obtain the measures before the release decision occurs. Some practices which help support this include:

Full-stack telemetry ▷ the ability to monitor for problems across the entire stack that a system covers

Visual displays ▷ tools that display automated measurements

Federated monitoring ▷ consolidated monitoring across applications in the solution that creates a holistic view of problems and performance

The [Measure & Grow](#) article provides some guidance on the types of metrics that monitoring requires.

Respond

The ability to respond to and recover from unforeseen production issues is critical to supporting continuous deployment and streamlining the CDP. The reasons are obvious:

Production issues directly affect customers and end users, so the value of deployed solutions can quickly erode when problems occur.

Production issues causes rework/fixes, patches, redevelopment, retesting, redeployment, etc. That disrupts the normal flow of value through the pipeline.

Since production issues can harm delivery efficiency and lower value, teams need the capability to detect problems proactively and recover quickly. As measured by Mean Time to Restore (MTTR), fast recovery is among the most reliable leading

indicators of high [DevOps](#) maturity [5]. Recovery is also one of the five elements of SAFe's [CALMR](#) approach to DevOps.

The goal of responding and recovering is to identify potential issues *before* they turn into incidents and to prevent them from affecting business operations. This capability requires detecting difficulties internally before end users discover them, quickly identifying root causes, and restoring services with well-rehearsed procedures. In contrast, making hasty, reactive changes directly to production systems—just to keep the lights on—invites source code and configuration differences between environments, unverified changes, and long-term risk.

Several practices support the ability to respond and recover from production issues:

Proactive detection △ a technique for proactively creating faults in the solution to identify potential problems and situations before they occur. For example, Chaos Monkey [1], developed by Netflix, is an open-source tool that randomly terminates instances in production to ensure that engineers implement their services to be resilient to instance failures.

Cross-team collaboration △ a mindset of cooperation across the [Value Stream](#) to identify and solve problems as they arise

Session replay △ the ability to replay end-user sessions to research incidents and identify problems

Rollback and fix forward △ the ability to both rollback a solution quickly to a previous environment or to fix a problem quickly through the pipeline without the need to rollback

Immutable infrastructure △ This concept refers to never amending servers or Virtual Machines (VMs) after deployment. Instead, a new server is built from an image with the appropriate changes if something needs updating.

Version control △ environments should be maintained under version control to rollback quickly

After teams have demonstrated that features have been deployed successfully to production and have the necessary monitoring and recovery capabilities to track and manage ongoing value, they have completed the *continuous deployment* stage of the CDP. In turn, this gives the enterprise the ability to release whenever warranted.

Enabling Continuous Deployment with DevOps

Continuous deployment involves critical operation activities frequently associated with the “Ops” in DevOps. They focus on deploying solutions to production environments, verifying their functional integrity, and ensuring effective monitoring

and post-release support.

Figure 3 illustrates that SAFe's CALMR approach to DevOps (center) and several practice domains (inner rings) enables continuous deployment. Each of the four activities (in green) is a collaborative effort that draws upon DevOps expertise from multiple disciplines to maximize delivery speed and quality.

Figure 3. DevOps enables continuous deployment

Figure 3. DevOps enables continuous deployment

For instance, *deploying* solutions in the CDP involves using tools that automate the provisioning of production infrastructure, deploy solution binaries to select targets, verify production functionality, capture runtime telemetry, and proactively alert on issues. DevOps practices and tools streamline these capabilities, allowing solutions to be deployed and fully prepared for on-demand release in minutes.

All four continuous deployment activities are enabled by DevOps, though with different combinations of technical practices and tooling. See the [DevOps](#) article

series for more guidance on DevOps and how it facilitates the CDP.

Learn More

[1] <https://netflix.github.io/chaosmonkey/>

[2] Kim, Gene, et al. *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. IT Revolution Press, 2013.

[3] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

[4] Humble, Jez, and David Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley, 2010.

[5] Gregory, Janet, and Lisa Crispin. *More Agile Testing: Learning Journeys for the Whole Team*. Addison-Wesley Signature Series (Cohn). Pearson Education, 2014.

[6] State of DevOps Report. <https://puppet.com/resources/whitepaper/state-of-devops-report>

Last update: 9 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



By multiplying milestones, we transform a long, amorphous race into one with many intermediate “finish lines.” As we push through each one, we experience a burst of pride as well as a jolt of energy to charge towards the next one.

—Chip and Dan Heath,
The Power of Moments

[Adjust Cookie Settings](#)

Facilitating SAFe Assessments

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

The Lean-Agile transformation and the journey to [Business Agility](#) is a significant undertaking for every [Enterprise](#). Many executives have commented that this transformation was one of the most difficult but most rewarding changes they have personally experienced in their careers.

The business benefits of business agility are clear: faster time to market for more innovative solutions; higher quality and productivity; higher levels of employee engagement; opportunity for a new and enhanced culture, and ultimately, the ability to thrive in the digital age. But for those new to the endeavor, the question arises about where and how to begin. The SAFe [Implementation Roadmap](#) provides a proven pattern of activities to adopt SAFe successfully. And yet, even when moving down the roadmap, the question for the enterprise becomes: How do we know how we are doing? Are we growing in the right areas? What do we do about the deficiencies we know we have? Where should we target our next effort?

To reinforce and accelerate the SAFe transformation, leaders need to Ômeasure and growÕ the implementation at various points along the journey. This will help maintain the energy and enthusiasm they are devoting to the short cycles of [Iterations](#) and [Pls](#) while setting their sights on the larger goals of true business agility. The SAFe assessments are key in identifying these improvement opportunities, and this article guides facilitating them successfully.

Details

ÔMeasure and growÕ is the term we use to describe how SAFe portfolios evaluate their progress toward business agility and determine the next improvement steps. It describes how to [measure](#) the current state of a portfolio and [grow](#) to improve overall business outcomes. The [Measure and Grow](#) article describes how this is achieved via three measurement domains: Flow, Outcomes, and Competency. This article focuses on the competency domain, measured via two separate assessment mechanisms designed for significantly different audiences and purposes.

1. The [SAFe Business Agility Assessment](#) (Figure 1) is designed for LPM and portfolio stakeholders to assess their overall progress on the ultimate goal of true business agility.

2. The *SAFe Core Competency Assessments* (Figure 2) help teams and trains improve on the technical and business practices they need to help the portfolio achieve that larger goal.

Each assessment follows a standard process pattern of *running the assessment*, *analyzing the results*, *taking action*, and, *celebrating the victories*.

SAFe Business Agility Assessment

The SAFe Business Agility Assessment is a high-level assessment that summarizes how Agile the business is at any point in time. The assessment report provides a visualization that shows progress measurements across the SAFe seven core competencies. An example report is shown below in Figure 1.

Figure 1. SAFe Business Agility Assessment

Figure 1. SAFe Business Agility Assessment

You can download the spreadsheet version of the assessment here.

[Download the SAFe Business Agility Assessment](#)

Note for SAFe Studio Members: All the SAFe assessments are available for SAFe Studio Members online through our partner Comparative Agility. This provides additional data collection, analysis, comparison, and trending capabilities that can be used to improve performance. Access these from the [Measure and Grow SAFe Studio page](#).

Running the Business Agility Assessment

Assessing business agility status is not a trivial feat. Opinions abound, the data is lumpy, and the ways of working are evolving simultaneously as the assessment is taking place. Therefore, simply sending the assessment out to various participants and asking them to fill in the data will probably not provide the right experience or accurate results. Instead, we recommend a facilitated session with someone trained in the nuances of SAFe and the assessment process. An experienced [SAFe Practice Consultant](#) (SPC) is probably a good choice.

Two assessment patterns can be used:

1. Each participant fills out the assessment independently, and then the group discusses and analyzes the results together
2. All participants discuss each statement together and reach a consensus on the score for each statement

Both patterns have their benefits and disadvantages. Trust the facilitator to pick the correct pattern based on group dynamics, distribution, and time frame.

Analyzing Business Agility Assessment Results

With the data from the assessment in hand, the next step is to analyze the results. During the analysis, it is essential to identify significant variances in opinion. The facilitator should review each area of disagreement and explore the differing views. These might stem from a different understanding of the statement itself or from disagreement about where the group is in the specific dimension. The goal is to explore the differences to get a better alignment of where improvement is needed. This is a significant part of the collaborative learning experience.

Core competencies that the group has assessed as problematic can then be explored to understand the reasons that drove people to score themselves low. In addition to pointing out areas needing improvement, the assessment allows portfolios to see visible improvements in performance or “wins.” The wins are small multiplying milestones that encourage teams to *consolidate those gains and produce more change*, as Kotter’s model [2] suggests.

The facilitator should also be aware of the Dunning-Kruger effect [3], in which people tend to assess their ability as greater than what it really is. This means that core competencies that seem unnaturally high might also require an examination to ensure the group understands the meaning of the statements in question.

Taking Action on the Business Agility Assessment

Although high level, taking the business agility assessment is, in itself, a learning experience. Many questions directly set expectations of behaviors, activities, or outcomes that can be reasoned about and discussed. For example, a question about continuous learning, such as “the organization provides time and space for innovation,” is relatively straightforward, and the implied corrective action is obvious.

Figure 1 shows the enterprise scored low in **Lean Portfolio Management** (LPM). That could be because they are ineffective at it, but it’s more probable that the enterprise hasn’t started that part of the journey yet. In most cases, a quick look at the implementation roadmap will identify some fairly obvious next steps, with the goal of steadily improving proficiency across all seven core competencies.

LPM or the **LACE** should routinely re-evaluate their progress toward business agility, perhaps every other PI, and plan the next steps. The measurement frequency depends on the opportunities pursued and how fast the portfolio can reasonably achieve progress. Creating a baseline early on in the transformation, followed by periodic assessments, will illustrate improvement trends and allow everyone to communicate successes.

SAFe Core Competency Assessments

In most cases, assessing progress toward business agility spurs the enterprise to greater and more profound efforts. That leads the business to explore and start to measure and take more specific action on some or all of the seven core competencies. Structured similarly to the business agility assessment, each core competency assessment has a set of statements, organized by dimension, rated on the same scale as the above. The questions go one step deeper to specific aspects

and areas of opportunity and concern along each of the three dimensions of that particular competency. An example report is illustrated in Figure 2.

Figure 2. A report from a team and technical agility competency assessment

Figure 2. A report from a team and technical agility competency assessment

Running a Core Competency Assessment

As with the business agility assessment, the scope, audience, and process for an individual competency assessment must be purpose-built. Low results in the Agile product delivery competency might require that each [Agile Release Train](#) (ART) in the portfolio assess its progress in that dimension. Or perhaps an LPM assessment needs to bring in the right stakeholders. In any case, all the guidance and caveats above apply, and attention to culture and careful facilitation is necessary to get the right experience and results.

(Note: These more detailed Core Competency assessments can be downloaded

from the bottom of this article.)

Analyzing Results of a Core Competency Assessment

The results of a competency assessment are summarized along three dimensions. But again, there is far more detail in the assessment, and far more learnings than the figure alone implies. For example, here's a sample of questions from the [Built-in Quality](#) dimension, which by themselves inform stakeholders and indicate improvement activities:

- Our team adheres to well-defined quality standards
- Our team practices both pairing and peer review
- Our team applies collective ownership to our work
- Our team's testing practices catch defects early

Taking the assessment, whether business agility or core competency, is not a mechanical effort. It's a fostered collaboration filled with learning, and it sets expectations and communicates intent. Therefore, even the simple act of taking the assessment will be a significant step towards improvement.

Additionally, it can be helpful to analyze the data in the following three ways:

Highest and lowest average scores: Highest average scores represent those areas where there is the greatest success. Identifying these can help to highlight the results of previous improvement efforts, and these strengths can be amplified further as required. The lowest average scores likely represent candidates for the next areas of improvement.

Most and least standard deviation: Often, the assessments will highlight differences in opinion. Comparing the standard deviation across the responses will illustrate where there is broad agreement on the progress being made and areas where there is disagreement. The latter warrants further investigation as it may point to siloed improvement efforts or challenges with communication or consistency of practice.

Comparison to a benchmark: One of the significant benefits of the assessments is that they can be used to show improvement trends over time. Comparing against a previous data set will immediately demonstrate whether our improvement efforts have successfully delivered the expected benefits.

To help manage WIP and focus improvement efforts, it's recommended to use this analysis to identify no more than five assessment statements that represent strengths and five statements that represent opportunities. The strengths will be amplified and celebrated, and the opportunities will be acted upon, as described in the next step.

Taking Action from a Core Competency Assessment

The next step is to identify the activities that will increase proficiency. We refer to these as growth recommendations. Collectively brainstorm growth recommendations for each statement that represents an opportunity, then affinity group and dot vote to get to 1-3 growth recommendations per statement (Figure 3). Rarely is it the case that a single growth recommendation can address all the challenges highlighted in an assessment statement, and therefore multiple activities will ensure that all facets are covered.

Finally, across all the assessment statements being addressed, these growth recommendations become potential backlog items that can be prioritized as described in the next step:

Figure 3. 3 growth recommendations created for a single assessment statement

Figure 3. Three growth recommendations created for a single assessment statement

Prioritizing Improvement Opportunities

To limit Work in Process (WIP) (see [SAFe Principle #6](#)) and ensure that something

does indeed get done, it is helpful to prioritize the opportunities and choose one or two that will provide the most value immediately. Like prioritizing features in the [ART Backlog](#) with Weighted Shortest Job First ([WSJF](#)), the same approach can be used to identify the best next growth opportunity to pursue.

A simple table to compare opportunities via WSJF is shown in Figure 4 below.

Figure 4. A sample spreadsheet for calculating WSJF for improvement actions

Figure 4. A sample spreadsheet for calculating WSJF for improvement actions

This approach will help the group select the improvement opportunities that yield the biggest impact with the least effort.

The prioritized opportunities go into the LACE backlog, the [Portfolio Backlog](#), or the ART backlog to be worked on as soon as possible. The backlog of choice depends on the opportunity. For example, an opportunity to restructure ARTs by running a Value Stream and ART Identification Workshop will likely be on the LACE backlog, while a recommendation to train all [Scrum Masters/Team Coaches](#) might belong in

the program or the portfolio backlog.

Core Competency Assessment Downloads

The following table provides download links for each of the core competency assessments.

Core competency assessments download

[Organizational Agility](#)

[Lean Portfolio Management](#)

[Enterprise Solution Delivery](#)

[Agile Product Delivery](#)

[Team and Technical Agility](#)

[Continuous Learning Culture](#)

[Lean-Agile Leadership](#)

Celebrate successes

Lastly, change is hard. Continuous change is more challenging. Intelligent enterprises use small wins to celebrate progress and inspire people to the next achievement milestone. There are many opportunities to celebrate: such as when a portfolio, ART, or team moves from one level to the next in each assessment dimension or perhaps even manages to change a single assessment statement from “mostly false” to “mostly true.” Celebrating successes creates the fuel needed for more improvement and advancement on the journey toward business agility.

These milestones can also provide an opportunity for organizations to gamify the business agility journey. This, in turn, can motivate individuals and teams to intensify their focus on the activities that will help them achieve their goals.

In addition, tying the improvement to changes in the [Value Stream KPIs](#) and [LPM metrics](#) connects the effort to the portfolio’s measures of overall success. In this way, the entire portfolio can focus on measurement and celebrate growth and positive outcomes.

Learn More

[1] Heath, Chip, and Dan Heath. *The Power of Moments: Why Certain Experiences Have Extraordinary Impact*. Simon & Schuster, 2017.

[2] Kotter, John P. *Leading Change*. Harvard Business Review Press, 2012.

[3] en.wikipedia.org/wiki/Dunning%20%93Kruger_effect

Last update: 24 October 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



To win in the marketplace you must first win in the workplace.

—Doug Conant, American businessman
and former CEO, Campbell Soup Company

Agile HR with SAFe

Bringing People Operations into the 21st

Century with Lean-Agile Values and Principles

[Adjust Cookie Settings](#)

by [Fabiola Eyholzer, CEO, Just Leading Solutions LLC](#)
with Dean Leffingwell, Co-founder and Chief Methodologist, Scaled Agile, Inc.

Note: This article is part of the [Community Contributions](#) series, which provides additional points of view and guidance based on the experiences and opinions of the extended SAFe [community of experts](#).

Introduction

The digital transformation is affecting virtually every enterprise across the globe. Competing requires a level of competence and ability in the development and deployment of software and systems unlike that which brought the successful [Enterprise](#) to this point. Such new competencies can no longer be mastered through industrial age structures and practice.

Responsive enterprises act by a) acknowledging talent, knowledge, and leadership as the new currency for competitiveness and by b) embracing Lean-Agile values, principles, and practices.

The Scaled Agile Framework ™ (SAFe) has emerged as the leading framework for addressing this challenge at enterprise scale. In addition to extensive practice guidance, SAFe promotes and describes a comprehensive set of values and principles that [Lean-Agile Mindset](#) leaders can use to foster the transformation and continue the journey to enhanced competitiveness in software and systems building. Figure 1 characterizes the SAFe approach.

Figure 1. The SAFe Lean-Agile Mindset

This new mindset challenges Human Resources (HR) to realign their people approach with this new way of working. It imposes a far-reaching transformation to bring HR into the 21st Century by shifting from a process-oriented HR Management to an empowering Lean-Agile people operations. It changes the face and significance of HR forever.

In this whitepaper, we will describe *six basic themes* that can guide Leaders and their HR Partners on how to address various aspects of more contemporary Lean-Agile people solutions in the Lean-Agile enterprise. These are:

- #1 ⚡ Embrace the New Talent Contract
- #2 ⚡ Foster Continuous Engagement
- #3 ⚡ Hire for Attitude and Cultural Fit
- #4 ⚡ Move to Iterative Performance Flow
- #5 ⚡ Take the Issue of Money off the Table
- #6 ⚡ Support Impactful Learning & Growth

#1 ⚡ Embrace the New Talent Contract

Today is the age of digital disruption and enterprises must respond to the new realities by fundamentally reinventing their mindsets, behavior, leadership, and ways of working. The driving force behind the Lean-Agile organizations is knowledge workers – the fastest growing and most critical workforce sector. Drucker defines knowledge workers as people who know more about the work they perform than their bosses. Their jobs consist of converting information into knowledge and instantiating that knowledge in systems and solutions. It is their ideas, experiences, and interpretations that keep businesses moving forward. The results of their intermediate work are often intangible and require improvisation; the use of judgment in ambiguous situations, as well as interactions with others, is continuously required. As such, it defies traditional, task-based management.

Knowledge workers thrive on this kind of challenging work. It motivates their very being. But they also seek meaning and purpose in their careers; and appreciation and respect. They want to take responsibility and be actively involved. In order to innovate and contribute, they must be allowed to manage themselves with significant autonomy and empowerment.

This forms the basis for creating and honoring the new talent contract. It is not only about understanding the drive of knowledge workers, but also recognizing the power shift that comes with it. In SAFe, this goes hand in hand with a move from task-management and command & control to inspiring leadership.

It is well elaborated in *two* of the [SAFe Lean-Agile Principles](#), specifically:

[Principle # 8 – Unlock the intrinsic motivation of knowledge workers](#)

[Principle # 9 – Decentralize Decision-Making](#)

This, inevitably, also affects the way HR interacts and engages with both management and the workforce. Employees will claim a voice in shaping the way their organization takes care of them – not only when it comes to their career development, but across the whole HR value chain.

Like management practices, *people operations* must become less prescriptive and more flexible, empowering and accommodating. HR solutions become co-created and evolve constantly. This is an integral part of building places of work full of inspiration and engagement.

#2 Foster Continuous Engagement

Tapping into the intrinsic motivation of people – keeping them deeply engaged in the enterprise’s purpose – has never been more important. Enterprises with

engaged employees have much higher returns. Yet the vast majority of employees worldwide are dissatisfied, disillusioned, and disengaged.

Agile understands the power of bringing intrinsically motivated people together to form collaborative empowered teams. In addition, SAFe then puts them onto an [Agile Release Train \(ART\)](#) where they are engaged with others on a common mission, and collaborate via face to face planning. Anyone who has ever participated in one of their [PI Planning](#) has firsthand experience of the enthusiasm and energy in the room. Unsurprisingly, SAFe teams are more passionate and involved.

Simply, engagement—perhaps sometimes dismissed as idealistic HR notion, translates directly into better business performance and success. Figure 2 illustrates the realities of disengagement and the benefits of high employee engagement.[1]

Figure 2. Employee Engagement

Engagement fosters retention. The best way to lower turnover is to invest in people. The concept of improving the market value of employees and making them more attractive for competitors may seem counterintuitive. But actively developing people takes away their need to switch jobs in order to improve and advance.

Agile practices allow people to evolve through challenging work, powerful collaboration, constant reflections, continuous feedback, and relentless improvement—all deeply embedded into the workflow. In other words, Agile does not distinguish between learning and working: Working equals learning. And knowledge workers are learning workers.

Hence, the goal of Agile Enterprises is not simply to retain talents, but to let them grow and thrive; and develop a flourishing talent pool.

#3 Hire for Attitude and Cultural Fit

Building a vigorous workforce starts with identifying, attracting, and hiring the right

people. But finding top talent is increasingly difficult. When it comes to Talent Acquisition, Agile Enterprises get a competitive edge by considering the following:

Build a strong employer brand ☐ Agile is a magnet for talented people. Enterprises can ☐ and should ☐ build on their commitment to Agile excellence and use it to help build a strong employer brand.

Proactively attract and engage knowledge worker talent ☐ It's a competitive market for top digital talent. Recruitment starts long before a new vacancy comes up. The talent acquisition team must continuously reach out and connect with interesting technical people to pull them into the talent pipeline.

Employ for attitude and cultural fit ☐ Agile is a team sport. Technical expertise is important, but Agile teams prosper when hiring candidates with the right attitude and cultural fit. The tendency for heroism and over specialization must be avoided. After all, success depends on the collective and collaborative skills of the team. Figure 3 shows some interview questions that can be used to address candidates' ability to thrive in a team setting[2].

Figure 3. Interview Questions

Inspire candidates with a larger sense of purpose ☐ Knowledge workers need a sense of purpose in their lives. Something beyond the bits and bytes, and even beyond their local team environment is required. Helping candidates understand the larger purpose of the organization is the best way to inspire them. Strengthen this by being genuine and dependable throughout the whole process.

Make a solid, team-based decision ☐ Talent acquisition is a shared responsibility, and no hiring decision should be made without the backing of the team. After all, no employee can thrive without team support. Therefore, the team must be actively included in the hiring process.

Excel at onboarding ☰ Agile has fairly well-defined team practices and roles and is unbeatable in getting new people up to speed once on board. Prior to that, however, pre-integration activities, interactions, and access to information can enhance the onboarding experience. Post onboarding support requires communication and touch points with the individual, to assure that they are happy in the role, and that management is content with their performance.

A high-quality recruitment process reduces the risk of bad hires and subsequent disruptions to the flow and performance of the team(s).

#4 Move to Iterative Performance Flow

Undeniably: Performance management is the most criticized HR process today. But despite a long list of complaints, many organizations still invest top dollars into a broken performance appraisal practice. The fact is, traditional performance appraisals simply don't work.

While Performance Management was initially installed to align goals and foster joint efforts ☰ it has since evolved into this the pivotal point for an entire set of HR practices, especially Compensation and Talent Management.

No wonder, annual performance reviews have turned into a tense time for everyone involved. Managers tend to batch feedback, both negative and positive, into the annual feedback dump, depriving employees of the timely feedback they need to actually improve when it matters; and employees are nervous about their appraisal as it affects their upcoming compensation and promotional chances.

Review tools are in place to help guide the discussion and ☰ force managers to rank their people. This means many companies excel at calculating ratings. But it comes at a high cost to the morale and motivation of people. However, the tides are shifting, and organizations of all shapes and sizes are already eliminating employee appraisals in order to respond to the challenges and interfaces to HR instruments. Figure 4 demonstrates the facts and trend of traditional performance management systems [3].

Figure 4. Employee Appraisals

This challenges HR to push the reset button on performance management on all levels. Here is how SAFe can help re-optimize the system for performance flow:

Align performance cycle in iterations -The accelerated pace of today's business world makes it increasingly difficult to think in multi-year periods and set rigid top-down annual goals. SAFe shifts from 12-month cycles to an iterative, interactive process measured in weeks and PI Plannings. These iterations reflect an optimal cadence and represent the new performance cycles.

Utilize PI planning to share the vision, set inspiring goals, and clarify expectations ▷ Being able to inspire people is imperative. The heartbeat of the ART is the PI Planning. All members of the train come together to understand the business context and vision, set and synchronize goals, clarify expectations and dependencies, and commit to common objectives. This replaces static individual goals with verifiable collaborative objectives.

Continuously inspect and adapt ▷ Agile ceremonies like [Iteration Reviews](#) and [Iteration Retrospectives](#) are all about inspecting and adapting. While iterations focus on the system under development, not the people, they do provide formal and informal opportunities in team retrospectives (each iteration) and [Inspect and Adapt](#) (program retrospectives) to provide feedback on individual and team performance. The focus is, hence, no longer on assessing individual goals, but on continuous improvement ▷ not only on a personal but also on a program enterprise level (something neglected in traditional performance appraisals).

Embed learning and development into the workflow ▷ Companies must constantly evolve, and Agile ways of working are all about learning. SAFe enhances it with the [Innovation & Planning \(IP\) Iteration](#) as an integral part of the PI. This gives people the time and space to improve and innovate at their own pace. Cross-role, cross-functional, and cross-team training further enhance the

skills, flexibility, and utility of the knowledge worker. These mechanisms support the particular need of the knowledge worker and help the enterprise fulfill its responsibility to create a workflow that allows for active learning and growth.

Eliminate annual performance ratings in favor of continuous feedback

Ø Instead of employee ratings, Agile organizations shape a culture of mutual respect where candid dialogues and continuous feedback consistently take place between leaders, employees, and peers. This focus on relentless improvement is a key part in embracing agility and achieving its full impact. It also fundamentally increases the intensity and quality of discussions on personal accomplishments and growth potentials.

These new ways of working fulfill the original goals of performance management far better than any annual performance rating. And by bringing a Lean-Agile mindset to all HR practices, people operations continuously engages, interacts, grows and recognizes talents Ø without the need for an employee appraisal to act as a trigger. Thus, making annual ratings and forced rankings obsolete in an agile world. Performance flow in Agile enterprises is all about optimal cadence, responsiveness, and growth Ø not about monetary incentives.

#5 Take the Issue of Money off the Table

The industrial era belief that money is the strongest (and only effective) motivator for employees is firmly rooted in many organizations. Unsurprisingly, compensation and cash bonuses are still used as a predominant way to incentivize and recognize people Ø an expensive and ineffective approach for knowledge workers.

Ever since Daniel Pink's "Drive" (and decades of scientific studies), it is clear: Agile people are driven by mastery, autonomy, and purpose. Meaningful reward and recognition for them come in forms of pride in achievement, social contacts, interesting work, new challenges, growth opportunities, and self-fulfillment.

The way to take the topic of money off the table is by paying employees fairly and competitively, but also by having open and honest communication about pay, and by making matching compensation with adequate reward and recognition.

Here are some guidelines to help rethink *reward solutions*:

Base Salary

Pay adequate base salaries Ø Roles in agile teams are based on generic value descriptions rather than specific individual job descriptions. Any job is

consequently less stringent and instead, requires a high degree of flexibility and ability to activate and apply relevant knowledge. Adequate base salaries, therefore, compensate not only the role but the person's skills and experiences. Once a knowledge worker accepts that they are fairly (and competitively) compensated, they are free to focus on the work, not the money.

Decentralize salary decisions ☐ Compensation has to be easy to deliver ☐ and change. This means empowering managers to set salaries and pay increases. These adjustments must be decoupled from an annual process. This allows for a more flexible schedule and helps separate the concern of compensation from performance feedback. Leaders are guided in their decision by peer reviews, transparency, as well as adequate data and expert advice, provided by a dedicated Reward Solution team as requested.

Bring transparency to the salary structure ☐ A transparent salary structure brings many advantages like fostering greater trust and honoring the value of employees, independent of their personal negotiation skills. (Indeed, many valuable knowledge workers are poor salary negotiators. In its place, they must feel that they are not disadvantaged by those who lobby more vociferously for money.) However, any company needs to evaluate (and test) carefully what degree of transparency is feasible in their environment.

Incentives

Avoid toxic individual bonuses ☐ Management-by-Objectives (MBO)-based individual, and team bonuses are toxic for an organization that thrives on collaboration and responsiveness. They set individual against individual and team against team. They should be eliminated entirely. However, fair, transparent incentives that honor collective performance and corporate success (for example, equity and profit sharing plans), allow employees to participate financially in the success of the enterprise and thereby feel they are fairly and justly treated for enterprise success. But avoid incentives that might provide an undesired motivation for people to stay or interfere with the company's need to move an employee out quickly.

Combine various forms of recognition ☐ Impactful appreciation must be aligned with corporate values. Each enterprise must find a suitable combination of low frequency formal, recognition with more frequent and intimate, personal acknowledgments. Once set, the power of recognition is put into everyone's hand. Figure 5 represents an example of a diversified recognition program [4].

Figure 5. Employee Appraisals

Benefits

Provide benefits that people value ☰ Benefits are more than pension plans and healthcare: they are about making the lives of people easier and better. Flexible schedules, remote office, parental leave, financial guidance, time off, and volunteer opportunities are examples of tangible benefits that help people find their optimal, work-life balance.

Invest in the health and well-being of people ☰ Companies face unnecessary waste in the form of stress, anxiety, burnouts, and chronic health issues. Investments in the well-being of employees are not only the right thing to do but translate directly into revenue. A balanced reward approach is a part of taking the topic of money off the table. It promotes a broader understanding of an effective incentive system ☰ one that appreciates the appeal of growth opportunities.

#6 Support Impactful Learning & Growth

Learning and growth have always been intertwined. But it takes on a completely new dynamic in a world of accelerating change, where ever shorter half-life of facts and knowledge is a reality.

Here are some points to guide the journey:

Create a learning organization ☰ While working is a key part of learning, agile people must also understand how knowledge grows, changes, and is overturned; and be given a method to acquire relevant new skills and competencies as well as the ability for transfer of learning. That is why a learning organization offers contemporary learning and teaching methods; captures and distributes intellectual capital, and provides transparency and access to knowledge and skills.

Empower employees to take the lead Ⓛ In line with new talent contract, employees are not only empowered when it comes to their work, but they are also in charge of their own development. They access knowledge, attend learning sessions, build a network, and shape their progress and career path according to their own needs and perceptions. This journey is strongly guided Ⓛ but not driven or controlled Ⓛ by people operations and Lean-Agile Leaders.

Illustrate prospective role-based career paths Ⓛ Modern careers are more about personal choices and meaningful growth than climbing a (fast disappearing) hierarchical ladder. Consequently, career paths are becoming more fluid, multifaceted, and individualized than ever before. A catalog of prospective role-based career paths can illustrate typical growth paths, but without limiting the options to a pre-set career model. Figure 6 presents an example of such a role-based career path [5].

Figure 6. Examples of role-based career paths

Establish individual career coaching Ⓛ A dedicated team of career coaches can connect with each person individually in order to outline an individual career profile with a learning and growth plan, that is continuously reviewed and adjusted as needed. This ensures that boosting the (internal) market value of people is always in focus, but it also equips the organization with a previously untapped understanding of their talent pipeline. HR no longer depends on a rating from an annual appraisal, because they know their people on a personal, authentic level.

Apply Agile workforce planning and talent scouting ☰ Allowing for flexible careers requires agile workforce planning and talent scouting. People operations must understand the organization and its changing needs and be able to match that with the circumstances and aspirations of people positively.

Enhance development through Lean-Agile Leaders ☰ Lean-Agile Leaders are lifelong learners, teachers, and people developers. They engage in continuous listening, communication, and feedback in order to identify development areas and boost learning. These constant interactions are crucial to the success of both: individuals and teams. The best leaders not only develop people but let them spread their wings ☰ even if it means letting them take on a new challenge in a different role/place within the organization. While it opens up an attractive world of growth opportunities, it also boosts skill sharing and knowledge transfer across the enterprise.

It is the aligned and combined effort of employees, leaders, and HR that enables the organization to become a learning network with an unbeatable talent pool ☰ the foundation of a resilient Lean-Agile Enterprise.

Summary

The face of organizations is changing fast ☰ and drastically. Lean-Agile development with SAFe reinvents the way we work and helps us build an engaged, talented, and vigorous workforce. It a valuable stepping stone to guide your HR into the 21st Century. But no matter where you are in your transformation journey towards a modern responsive Enterprise, the time is right to embrace Lean-Agile People Operations and become equipped to deal with the challenges of today's organizations and people.

So, Agile Professionals: Reach out to your HR organization and invite them to SAFe training, ceremonies, and learning sessions; and support them in bringing a Lean-Agile mindset to their People approach.

Likewise, HR Professionals: connect with Agile teams and experience firsthand the power of Agile; and become educated and knowledgeable about Lean-Agile values, principles, and practices.

Investing in Lean-Agile people operations is an investment in your people and your future.

Download

Learn More

[1] Sources: Gallup via Employee Engagement & Loyalty Statistics 2014 by B. Carter, Office Vibe 13 Disturbing Facts About Employee Engagement, November-2014, American Management Association (AMA) Database 2015 D Torben Rick, Infographic 2014, Employee Engagement 2014, Daily Infographic, Scaled Agile Case Study Meets Big Iron at John Deere

[2] Just Leading Solutions, 2016.

[3] Sources: Corporate Executive Board 2014 | SHRM Survey: HR Professionals' Perceptions about PM effectiveness (Oct-21-2014) | Harvard Business Review April 2015: Reinventing Performance Management by Marcus Buckingham & Ashley Goodall | The Washington Post Jul-21-2015: In big move, Accenture will get rid of annual performance reviews and rankings by Lillian Cunningham | Co.Tribute: A Performance Review That Actually Means Something by Carson Leith, Mar-2016 | Harvard Business Review, June, 2012; Talent Management, May 2012 Image, Discovery Education | Infographic 11 Eye-Opening Statistics on the Importance of Employee Feedback, Officevibe 2015 D Why Employee Performance Reviews Are So Old School, Millie Dent, The Fiscal Times, Jul-2015.

[4] Adapted from Nokia New Recognition Framework, HR Tech World Congress, 2015.

[5] Just Leading Solutions, 2016

Last Update: 3 January 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

PARTNER

[Becoming a Partner](#)

CONTENT &
TRADEMARKS

SCALED AGILE,
INC

[Posters & Graphics](#)

[Watch and download](#)

[SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework

Agile Workspaces

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Adopting Lean and Agile practices across the Enterprise greatly benefits from a physical environment for teams and [Agile Release Trains](#) (ARTs) that support these new working methods. Experience and research have also shown that effective workspaces are one of the keys to high productivity. Further, the environment must positively reinforce the cultural changes while amplifying the focus on face-to-face communication and information transparency.

[Adjust Cookie Settings](#)

We describe the following physical spaces that encourage collaboration, communication, and innovation as *Agile Workspaces*:

1. **Workspaces for Agile teams** that balance the need to work together daily alongside occasional privacy
2. **Cross-Team collaboration spaces** that support critical events such as PI Planning as well as encourage innovation
3. **Work visualization techniques** that create transparency and alignment alongside opportunities to improve processes

Workspaces for Agile Teams

In the search for ever higher productivity and employee engagement, many enterprises have experimented with team workspaces explicitly designed to support Agile development[1]. Such experiments are not efficiently conducted, as arranging and rearranging a physical workspace involves facilities, capital expense, potential interior remodels, and more. Out of these experiences, a few standard requirements have emerged:

- Provide individual focus space for being *in the zone* of developing code, tests, and other artifacts
- Support the need for constant informal team collaboration
- Support the need for occasional privacy
- Provide room for the Team Sync and space for whiteboards, visual information radiators, and at least one large monitor

We've observed practices at the extremes, from individuals in traditional cubicles to completely open workspaces designed for ad hoc pairing, teaming, and ease of reconfiguration. While opinions vary considerably (XP shops, in particular, appear to favor the fully-open pairing concept), in our experience, neither extreme is optimal. Many organizations have arrived at a reasonable standard and efficient design, as illustrated in Figure 1.



Figure 1. An Agile pod, augmented, where feasible, by a team room and private phone cubicles

In this design, teams of 8-10 individuals work in a "pod" of semi-private cubicles. The walls within the pod are low, allowing for informal discussions. Their focus space is inside their cubicle, yet it will enable informal communication and knowledge gain from hearing the team members' conversations. However, the higher outside walls of the pod essentially separate the noise and conversation from the team next door. In addition, the gathering spaces in the middle provide an area for hoteling team members, spontaneous pairing, and quick, informal sessions. The pod is supported by whiteboards, a large monitor for video conferencing, and information radiators.

Ideally, a separate shared meeting room with video conferencing and private phone cubicles should be near the pod, as illustrated on the right of Figure 1. The meeting room allows the team to post their information on the walls and collaborate with remote team members and other teams. When not used by the teams, it also provides a retreat for private conversations. Sometimes, these team rooms can be shared with other teams and serve as an informal conference rooms

for others. Where feasible, a few personal phone cubicles support the need for individual privacy. In some cases, these have been provisioned as movable and stand-alone Ôphone boothsÕ inside a more open workspace.

Workspaces for Remote Workers

Agile development was initially designed and optimized for collocated teams. After all, Ôthe most efficient and effective method of conveying information to and within a development team is a face-to-face conversationÓ [2]. To this end, enterprises often spend significant time and energy establishing collocation. The Agile pod design above is a reflection of that.

However, collocation is not always feasible, especially during the pandemic and now in the current COVID-19 endemic. Some people who can contribute the most cannot be locatedÑor relocatedÑon-site. Today, many Agile teams are distributed across geographies, sometimes with significant time zone differences.

We have both witnessed and participated in high-performing teams with a significant degree of distribution. While being distributed does not change the basic Agile ways of working, it does impose some requirements on the business and the remote workers.

- High bandwidth video and audio connectivity
- Tooling for the team and ART Kanban boards and backlogs
- A wiki or intranet site providing access to [Strategic Themes](#), [Portfolio Vision](#), and other critical information
- Collaboration tools for communication, visualization, and ideation
- Overlapping core hours for the Team Sync, [Iteration Planning](#), demos, and other events
- Commitment to routinely travel to [PI Planning](#) events

Cross-Team Collaboration Spaces

In our experience, successful cross-team collaboration must support two work patterns. The first is to bring the entire ART together for critical SAFe events on a predictable cadence. These include PI planning, [System Demos](#), and [Inspect and Adapt](#) (I&A) events. Additionally, individuals need space to collaborate across teams to innovate away from the pressures of their day-to-day work, generally in a workshop-style format when needed.

ART Collaboration Spaces

PI planning is one of the most critical events SAFe, and a semi-dedicated physical

space for planning is prudent and will pay for itself over time. In addition to the physical space, adequate communication channels must be established for attendees who cannot be in person. For many enterprises, room for a single ART or two is about all that can be allocated permanently. Figure 2 provides a typical room layout for a PI Planning event.

Figure 2. A typical room layout for a PI planning event

Figure 2. A typical room layout for a PI planning event

One memorable anecdote is the management help desk. This emerged organically in one enterprise when it recognized that it was impractical for all the Business Owners to attend both full-day sessions of each PI. Their workaround was to create a help desk where one or more **Business Owners** were always present and who could reach others on demand for real-time decision-making. (Note: We think this help desk triggered one of the cultural turning points in that enterprise).

In larger multi-ART implementations, it may be impractical to provide a dedicated space large enough for all the ARTs. In this case, a single on-site ART planning location becomes the cornerstone but facilitates are extended during PI planning to include hotel conference centers and other venues.

Of course, this space won't be used just for the regular PI planning schedule. During the PI, it will also host the system demo, the I&A, and additional team-sharing spaces. (This is covered in more detail below.) Alongside cadence and synchronization, a permanent location adds predictability, minimizes uncertainty, and reduces the transaction cost of organizing these events. In our experience, this space adds to the collaborative workspace strategy that benefits the Agile enterprise.

Providing Space for Innovation

Innovation is a different and inherently creative process apart from current roadmaps and features. To achieve real innovation, people need time to put aside their daily responsibilities and dwell on the more significant problems and opportunities. This need encourages more extended periods of backlog-free time to explore potential ideas and prototype and test those ideas with your current customers.

The physical environment plays an important role here, not only in creating a collaborative space away from the daily work. It also supports the variety of approaches used during the creative process. Success patterns that we have observed include the following:

Ease of booking rooms ▶ To ensure they are available for running creative and collaborative workshops, these rooms are prevented from being block-booked for recurring events. Excessively using these spaces for less critical purposes leads to delays in collaboration and innovation or teams giving up on trying to use the area.

Facilitation kits ▶ Both inside this room, but also across the organization, facilitation kits should be readily available. These generally include pens, stickies, scissors, tape, and so on to support any innovation process. A working agreement ensures that the teams restock the kits and return them once finished, reducing the preparation overhead for the next team.

Moveable whiteboards ▶ Although fixed whiteboards in each room are beneficial for illustrating ideas and capturing information, they must be cleaned when vacating. A better solution is moveable whiteboards that are wheeled in, then kept for a while, ensuring that the output is not lost and can be worked on in future sessions.

Rearrangeable furniture ▶ The room's layout must adapt as the event's purpose changes. Avoiding fixed furniture will allow people to set up these spaces for various scenarios, such as multiple tables for group work, a lecture-style arrangement to present ideas, or removing the chairs and tables to create a completely open space.

Video Conferencing Capabilities ▶ Often, cross-team collaboration involves a few individuals who cannot attend in person. Always-on video conferencing capabilities in these spaces ensure everyone can participate. In addition, the use of robots, which

the remote attendees can control to move around the room, enhances their inclusion.

Visualizing Work

During various SAFe training forums, attendees frequently ask, “but if I was to leave here and do only one thing to start implementing Lean-Agile development at scale, what would it be?” Our answer is always the same: *Visualize the work*. That is why when you visit an Agile enterprise, you **see** work everywhere—on walls, whiteboards, monitors, wherever you look. Visualization converts the intangible to the tangible, flushes out unnecessary, unplanned, unapproved, or duplicate work, and aligns everyone to the current state instead of simply following the plan. The common thread across all these approaches is that the information is always available without the effort to discover it.

Based on our experiences, we’ve recommended some starting points below.

Visualizing Who We are Working For

Agile development is obsessively focused on the customer; that’s one of the things we like most about it. To ensure they are addressing the problems of real customers, Agile Teams use personas to bring the customer to life. (See [Design Thinking](#)). It’s common to see even life-size cardboard cutouts of one or more personas in the team workspace. Along with the documented attributes of the persona, these serve as constant reminders of who the teams work for and how they support the jobs to be done by their real user.

Visualizing the Flow of Work

Lean teaches us that you can’t manage what you can’t see. Making the current work visible exposes the amount of work in process, bottlenecks, and what people are doing instead of what we think they are doing. It highlights the status of the flow of value throughout the system. A general-purpose Kanban system can manage any work following this approach. The left and right sides of Figure 3 show the Portfolio and ART Kanban systems we use at Scaled Agile to manage [Epic](#) and [Feature](#) flow.

Figure 3. Managing the flow of work with a Kanban system and ART planning board

In addition to monitoring the progress of work items through the system, dependencies between teams also warrant close attention. The ART planning board, also visible in Figure 3 and created during PI planning, captures this information and should be displayed prominently during the PI. Successful ARTs use the ART planning board as a focal point for Coach Sync events.

Visualizing Strategy

Figure 4 illustrates another example of visualizing work. This image shows an investment corridor identifying all current and potential epics in flight at one large enterprise. This corridor was the planning visualization for a PI-like, large-scale portfolio prioritization workshop.

Rather than confining it to a room, information is readily visible, making it easy for people to walk up and add their thoughts and suggestions. Also, making the transparency of data brings the strategy into the open and helps create alignment across the organization. Other examples we have seen often include the additions of the enterprise and portfolio's strategic themes, vision, and [Roadmap](#), to give a complete and rounded picture of the current and future direction of the business.

Figure 4. The investment corridor for a portfolio prioritization workshop.

Visualizing Solution Health

Customer support teams have long seen the value in displaying the number of waiting calls, daily closed and open tickets, and current SLA levels prominently on monitors close to the teams who rely on that information. Agile teams have

adopted this approach to include metrics on the state of the current solution. Examples of these include critical quality metrics, such as:

- Build health
- Number of defects
- Percentage of passing tests
- Performance and load of the solution infrastructure

An example can be seen in Figure 5.

Figure 5. Monitors displaying build health, test status, and current application usage

This real-time information needs appropriate tooling, which collects data the solutions provide. (For more information, see the [DevOps](#) article.) It's a worthwhile investment as it visualizes the association between cause and effect and helps teams quickly assess the impact of frequent application changes, leading to faster resolution of issues. The public availability of this information creates a healthy balance of accountability alongside a constant drive for relentless improvement.

Summary

This article has shared how successful enterprises have adapted their physical environments. All these changes have one thing in common: to remove the barriers to adopting Lean-Agile working methods. Although making these changes can involve an initial investment, the business benefits delivered are substantial and make it worthwhile.

Agile workspaces create the conditions for high-performing teams to emerge
Space for cross-team collaboration and innovation ensures that we deliver on our short- and longer-term business objectives
Visualizing the work brings opportunities for process improvement into focus and accelerates value delivery

In today's endemic, enterprises are at a crossroads, determining if work will be performed in-person, remotely, or in a hybrid environment. In any case, many of the concepts in this article can apply to all three work scenarios. However, they must be adapted to hybrid and remote environments. (See the [Working Successfully in Agile with Remote Team Members.](#))

Learn More

[1] Hesselberg, Jorgen. *Unlocking Agility: An Insider's Guide to Agile Enterprise Transformation*. Addison-Wesley Signature Series (Cohn), Pearson Education.

[2] Manifesto for Agile Software Development. <https://agilemanifesto.org>.

Last update: 22 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

SCALED AGILE, INC

[CONTACT US](#)

[SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[Permissions Form](#)

[Usage and Permissions](#)

5400 Airport Blvd., Suite

300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



To improve entrepreneurial outcomes and hold innovators accountable, we need to focus on the boring stuff: how to measure progress, how to set up milestones, and how to prioritize work. This requires a new kind of accounting designed for startups— and the people who hold them accountable.

—Eric Ries, *The Lean Startup* [1]

Adjust Cookie Settings

Applied Innovation Accounting in SAFe

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

Introduction

Developing innovative world-class solutions is an inherently risky and uncertain process. But this level of uncertainty causes some enterprises to avoid taking prudent risks. When they do, it increases the likelihood they will spend too much time and money building the wrong thing based on flawed data or invalid assumptions. As Eric Ries observed, “What if we found ourselves building something nobody wanted? In that case, what mattered if we did it on time and within budget?” [2]

Unfortunately, traditional financial and accounting metrics have not evolved to address the need to support investments in innovation and business agility. As a result, organizations often use *lagging* economic indicators such as profit and loss (P&L) and return on investment (ROI) to measure the progress of their technology investments.

While these are helpful rear-view mirror business measures, these results occur far too late in the solution lifecycle to inform the actual solution *development*. Even NPV (Net Present Value) and IRR (Internal Rate of Return), though more forward-looking, are based on estimating the unknowable future cash returns and speculative assumptions of investment costs and discount rates. Moreover, even with a sensitivity analysis (such as “what if” calculations), these financial metrics neither reflect nor inform the learning from incremental product development. Consequently, these traditional metrics aren’t helpful in our move to iterative, incremental delivery and [Business Agility](#).

We need a better plan that quickly validates product assumptions and increases learning. In SAFe, this is accomplished, in part, by applying the *Lean Startup Cycle* (see Figure 1), a highly iterative process that provides the opportunity to quickly evaluate large initiatives and measure viability using a different financial measure: *innovation accounting*.

Figure 1. Lean Startup cycle with SAFe

Figure 1. Lean Startup cycle with SAFe

What is Innovation Accounting?

Innovation accounting is a term coined by Eric Ries' book, *The Lean Startup* [2]. This process cycle consists of 3 *learning milestones*:

1. **Minimum Viable Product** (MVP) → establish a baseline to test assumptions and gather objective data.
2. **Tune the Engine** → quickly adjust and move towards the goal based on the data gathered.
3. **Pivot or persevere** → Decide to deliver additional value or move on to something more valuable based on the validated learning. (Lean thinking defines value as providing benefit to the customer; anything else is waste.)[2]

A fast feedback loop is essential to validate learning and reduce waste, also called "build-measure-learn," as illustrated in Figure 2 [2]. Applying the knowledge obtained from customer feedback results in increased predictability, decreased

waste, and improved shareholder value.

Figure 2. Build, Measure, and Learn feedback loop from Lean Startup [2]

Leading Indicators versus Vanity Metrics

Innovation accounting asks us to consider two questions:

1. Are we progressing toward our outcome hypothesis?
2. How do we know?

In *The Lean Startup*, this is known as a *leap of faith assumption*, which requires that we understand and validate our value and growth hypothesis before we move forward with execution. This becomes essential to the economic framework that drives effective solution development.

To answer these questions and make better economic decisions, we support innovation accounting using *leading indicators* and actionable metrics focused on measuring specific early outcomes using objective data. Leading indicators are designed to harvest the results of developing and releasing a Minimum Viable Product (MVP). These indicators may include non-standard financial metrics such as active users, hours on a website, revenue per user, net promoter score, and more.

It's essential to be aware of vanity metrics, which are indicators that do not truly measure the potential success or failure of the actual value of an initiative. While they may be easy to collect and manipulate, they do not

necessarily provide insights into how the customer will use the product or service. Measures such as the number of registered users, raw page views, and downloads may provide helpful information or make us feel good about our development efforts. Still, they may be insufficient to provide the evidence to decide if we should pivot or persevere with the MVP.

There are some practical ways we can avoid being deceived by vanity metrics and instead work to evaluate our hypothesis. A/B or split testing enables us to validate our outcome hypothesis using actionable data.

For example, group A may get the new feature, and group B does not. By establishing a control group, we can evaluate the results against our hypothesis and make decisions as part of our feedback loop. We can also avoid vanity metrics by focusing on customer-driven data.

We can also use *cohort analysis* to examine the use of a new product, service, feature, and so on over time as it pertains to a cohort (group). For example, suppose we wanted to see how a new feature on our website improved the conversion rate to paying customers. We could look at new weekly registrations of the cohort and report on the percentage conversion to paying customers. We can analyze this information weekly and see if each cohort's conversion rate remains constant. If it does, we have a clear indication of how the feature is affecting the conversion rate. If it does not stay constant, then we have a chance to tune the engine or pivot.

Applying Innovation Accounting in SAFe Measuring Epics

Implementing large, future-looking initiatives is an opportunity for organizations to reduce waste and improve economic outcomes. In SAFe, large initiatives are represented as *Epics* and are captured using an *epic hypothesis statement*. This tool defines the initiative, expected benefit outcomes, and the leading indicators to validate progress toward its hypothesis.

Example of Airline Website Epic

For example, consider an airline that wants to develop a website for purchasing tickets. This is a significant endeavor that will consume considerable time and money. Before attempting to design and build the entire initiative, the epic hypothesis statement template should be used to develop a business outcomes hypothesis, define leading indicators and gain knowledge regarding the expected outcome (see Figure 3).

Figure 3. Epic hypothesis statement

Figure 3. Epic hypothesis statement

We might hypothesize that the website will help reduce call center volume and ultimately reduce costs to the airline, resulting in better profit margins per ticket sold. Thus, we are assuming that the website will be faster and easier than a phone call to the customer service department.

To test that hypothesis, we could release an incremental feature or set of features, such as an MVP, that allows customers to research flight schedules. We could analyze the call volume and the types of questions the help desk received to validate and measure the feature's effectiveness. Then we could quickly compare the trends of inquiries on the website vs. those at the call center. Additionally, we could build telemetry for collecting data into the feature using DevOps practices and analyzing customer interaction. The information captured in Figure 4 shows that call center activity has *decreased*, and website use has *increased*. These leading indicators demonstrate that the MVP appears to validate our hypothesis.

Figure 4. Call center telemetry

Figure 4. Call center telemetry

Leading indicators can provide immediate feedback on usage patterns. By analyzing the objective data, we can test our hypothesis and decide to continue releasing additional features, *tune the engine*, or pivot to something else. Thus, the Epic's MVP and the leading indicators enable us to make faster economic decisions based on facts and findings. Interestingly, in figure 4, the *visits to the site* metric by itself might be considered a vanity metric. This metric doesn't tell us much about our MVP's success or Epic's viability. However, placed in context with the other metrics, it indicates where customers spend their time visiting the website.

Example of Autonomous Vehicle Epic

With their direct connection to large numbers of users and their interactions, as well as the ability to quickly evolve the UI, websites are a convenient place to consider how to apply innovation accounting. However, it has far broader applicability than that. Let's consider a different example, an epic, which describes

the sensor system of a new autonomous vehicle.

Our epic hypothesis (see Figure 5) is that the sensor system will quickly detect and help us to avoid collisions with objects. We assume the information will be provided fast enough for the vehicle to stop or take evasive action (as that is the entire point!).

Figure 5. Example Epic for an Autonomous Vehicle

Figure 5. Example Epic for an Autonomous Vehicle

To test the hypothesis, we would like to find out if the sensor system can detect objects and if it's fast enough for our purposes *before* building the entire system. We could make a single sensor and essential data capture software to validate the distance between the object and the speed of the vehicle control system interface. As an MVP, suppose we mounted the sensor on the front of a car and connected it to a laptop within the vehicle.

Next, we place several objects on a test track and drive the car toward them. We could use the software to record information from the sensor as our leading indicator. We could also use software to measure how long it takes for the

message to be generated and sent to the vehicle control system interface by using a mock-up instead of waiting for the vehicle control system to be built. This would give us an early indication of compliance with our NFR. Figure 6 describes the leading indicators for this cyber-physical system.

Figure 6. Leading Indicators for autonomous vehicle sensor

Figure 6. Leading Indicators for autonomous vehicle sensor

Based on the data generated during the MVP testing, we have more questions to answer before moving forward with additional features. Perhaps we can *tune the engine* and see if we can decrease the message-sending time. Why didn't the sensor detect the road sign? Would going slower have helped for the initial tests? What happens if we place the sensor on the roof of the car? We may need to pivot to a different technology based on the answer to these and other questions.

ÔFailing fastÕ is an Agile value. The implication is that failure in small batch sizes is acceptable *if we learn from it*. Thus, validated learning becomes the primary objective of testing the hypothesis. As previously mentioned, SAFe uses the Lean Startup Cycle to evaluate the MVP of Epics iteratively and to *pivot* (change direction, not strategy) or persevere (keep moving forward) decisions against the

outcomes hypothesis. This is done incrementally by developing and evaluating features from Epic. We use the empirical metric to prioritize features by applying Weighted Shortest Job First ([WSJF](#)). With WSJF, we can rapidly assess the feature's economic value and epic's overall progress toward its hypothesis. This allows us to quickly and iteratively make a pivot-or-persevere decision based on objective data.

The decision to persevere indicates there is still additional economic benefit. Leading indicators validate our hypothesis and MVP, resulting in the development and release of other features. The decision to pivot may occur when sufficient value has been delivered or upon learning that our hypothesis was incorrect. Deciding to pivot is so difficult that many companies fail to do it.^[2] Often companies will continue to invest in an initiative despite (or lack of) data to the contrary. We can reduce the waste of time and money by using fast feedback loops and leading indicators to avoid working on features that customers don't want or need.

It's important to note that innovation accounting, as applied to SAFe, does not consider the *sunk cost* (such as money already invested). To pivot without mercy or guilt, we must ignore sunk cost, as discussed in [SAFe Principle #1 - Take an economic view](#). Although [Lean Budgeting](#) may allocate funding to a value stream upfront, we continually use the Lean Startup Cycle to evaluate the benefit hypothesis. Consequently, the initial *allocation of funds* does not equate to actual *spending*, so deciding to pivot or preserve is crucial.

Summary

Variability and risk are part of every significant technology initiative. However, traditional financial metrics used to measure the value of those initiatives *during* development have not evolved to address the need for innovation and business agility. Innovation accounting was developed as part of the Lean Startup Cycle to provide validated learning and reduce waste. Creating and measuring an MVP is used to validate the hypothesis, obtain results, and minimize the risk before investing in the entire system. This fast feedback loop, based on objective data and actionable metrics, enables incremental learning. Focusing on the right leading indicators ensures we make the vital pivot or preserve decision. SAFe's use of innovation accounting and the Lean Startup Cycle enables the enterprise to reduce waste and accelerate learning while enhancing business outcomes.

Learn More

[1] <http://knowledge.wharton.upenn.edu/article/eric-ries-on-the-lean-startup/>

[2] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. The Crown Publishing Group, 2011.

Last update: 24 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

[Posters & Graphics](#)

Watch and download
SAFe videos and
presentations

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



A long-term relationship between purchaser and supplier is necessary for best economy.

ÑW. Edwards Deming

Supplier

A Supplier is an internal or external organization that develops and delivers components, subsystems, or services to ARTs and development value streams.

Lean-Agile [Enterprises](#) deliver value to [Customers](#) in the shortest possible lead time with the highest possible quality. Suppliers support this mission by providing their unique competencies, skills, and existing [Solutions](#) to accelerate and reduce the costs of value delivery. Many [Solution Trains](#) and [Agile Release Trains](#) (ARTs) depend on supplier performance for their value delivery.

Suppliers will use varying development and delivery methods. Nonetheless, the SAFe enterprise treats strategic suppliers as long-term business partners, involving them deeply in the solution's definition and building activities. They also work with suppliers to help them adopt [Lean-Agile Mindsets](#) and practices for the economic benefit of both parties.

Details

Suppliers play a critical role in SAFe. They provide unique expertise and existing components that accelerate solution delivery. Some suppliers are external to the enterprise. Others work within the organization, producing solutions as part of another value stream. Suppliers will have their own mission, additional customers, and [Economic Framework](#) that drive their decision-making processes. For organizations to achieve mutual benefits, close collaboration and trust are required.

Types of Suppliers

Supplier relationships are as varied as the suppliers, and enterprises have many types of contractual associations with external entities. Some common relationship patterns include:

Licensing commodity components and services. Some suppliers provide commodity solutions. For example, web developers build their applications on platforms like WordPress and utilize third-party services to authenticate email addresses and process credit cards. When working with these suppliers, solution builders should maintain alignment between their [Solution Roadmap](#) and their suppliers' roadmaps and release schedules.

Contracting individuals and teams. Some suppliers provide skills vital to delivery, ranging from domain and technical expertise to the [SPC](#) and [Team Coach](#) roles that support a Lean-Agile, SAFe transformation. These suppliers should operate like other SAFe practitioners on teams and trains.

Contracting an entire development effort. Some organizations may outsource all development for a solution to a supplier who provides Agile teams, product

management, architectural support, and other expertise. Ideally, these suppliers operate harmoniously with SAFe practices, with the organization acting as [Business Owners](#) to guide the development effort.

As illustrated in the SAFe Big Picture, a fourth type of supplier provides strategic, customized solutions, requiring co-development.

Co-developing a solution ☰ Co-development suppliers provide strategic, customized solutions that require collaborative development and frequent integration with the overall solution. Their participation is necessary for the success of the larger solution and must be integrated to support more complex and strategic contributions than the other supplier models. Specifically, they contribute customized code, solutions, subsystems, and other assets.

The remainder of this article describes building an effective relationship with these critical co-development suppliers.

What is a Supplier in SAFe?

Suppliers in SAFe provide solutions and services to a SAFe [Development Value Stream](#) (DVS) to integrate into a larger solution, as shown in Figure 1. The direct customer is the DVS consuming the supplier's work (see Customer definition in the [Customer-Centricity article](#)). Together, the direct customer and supplier define and evolve the technical interface described in the [Solution Context](#). They also establish how the DVS will consume the supplier's changes as their solution evolves, as shown by the direct customer's [Operational Value Stream](#) (OVS). For example, suppliers may directly integrate their changes into the DVS's Continuous Delivery Pipeline (CDP). Or they may periodically package their solution and provide it to the DVS developers for them to integrate. Together, they must address installation, support, training, and other OVS concerns.

Figure 1. Suppliers deliver value to the overall solution

Figure 1. Suppliers deliver value to the overall solution

Figure 1 also shows how suppliers have customers at multiple levels. In addition to meeting their direct customer's needs, suppliers must also understand how the end-user, or indirect customer, employs their solution. Suppliers apply [Design-Thinking](#) to understand the needs of all critical customers.

Selecting Suppliers

When contracting with vendors and suppliers, organizations sometimes emphasize near-term pricing at the expense of longer-term economics. There are other anti-patterns as well:

Organizations searching for the lowest price may change suppliers without understanding the indirect costs of change, including lost knowledge and relationships

Organizations may aim to reduce or limit their number of suppliers without

recognizing their unique strategic advantages

Instead of genuinely leveraging what a supplier can offer, organizations may assign work using detailed, mandatory specifications, missing the opportunity to leverage a supplier's innovations

Due to confidentiality and other concerns, organizations may isolate suppliers from the enterprise's goals and strategies, providing information only on a need-to-know basis

Contract structures may limit a supplier's ability to adapt, committing them to early, predetermined requirements and a fixed, detailed schedule

While protecting investments is admirable, developing innovative products with fast time-to-market requires a different approach to supplier relationships that addresses the inherent technical, market, and business uncertainty the enterprise faces.

To enable business agility, Lean-Agile enterprises need a longer-term economic perspective, cultivating a collaborative, ongoing, and trusted relationship with their strategic suppliers. These suppliers become an extension of the culture and ethos of the enterprise; they are treated as partners. Their capabilities, policies, and economics are surfaced and understood. Strategic suppliers are selected based on their alignment in multiple dimensions:

Technical and business-model fit with the overall solution

A development model consistent with the buyer's way of working

Cultural alignment with the buyer's purpose, both now and in the future

However, reaching this state can be challenging if the supplier's mindset, philosophy, and development approach are materially different from the buyer's. Consider two cases:

1. The supplier has already embraced and adopted Lean-Agile development.
2. The supplier operates in the traditional waterfall model.

Typically, larger enterprises must address both, but the goal is the same—a more collaborative, long-term, and transparent partnership. The following two sections describe these two models.

Working with Lean-Agile Suppliers

Involving suppliers with a Lean-Agile mindset and existing ways of working is the more straightforward case. Suppliers already operating in SAFe are ideal since the terminology and development models are already aligned. In these instances, organizations can apply Lean-Agile collaboration practices:

The supplier is treated like an Agile Release Train (ART) and works in the same cadence as the other ARTs

The supplier participates in [Pre-Plan](#) and [Coordinate and Deliver](#) activities, including [PI Planning](#)

Their dependencies with other ARTs appear on the [Solution Train Planning Board](#)

The supplier demos their subsystem or components in the [System Demo](#), participates in the [Solution Demo](#), and continually integrates their work, providing feedback and receiving it from other ARTs and teams

The supplier participates in [Inspect and Adapt](#) (I&A) to improve their Lean-Agile practices and enhance the entire development value stream

Working with Suppliers Using Traditional Methodologies

Occasionally, systems builders still need to work with suppliers using traditional methods. Even in these cases, continuous collaboration across the organizations provides significant value and should inform the selection process discussed earlier. The more strategic the relationship, the more that alignment becomes critical to success. However, when working with traditional suppliers, organizations may need to make some adjustments:

Since the supplier will expect more formal requirements and design documents, System Architects and Product Management must allocate time for themselves and certain teams to create and evolve the specifications

The supplier may not be able to deliver incrementally; frequent integration may be challenging

Changes to requirements and designs need to be understood earlier, and the response to changes will take more time

Organizations should also set some expectations:

Suppliers should maintain alignment with the solution roadmap as it evolves.

Suppliers should communicate their progress toward upcoming milestones in Pre-Plan activities to allow dependent DVSs to plan properly.

Key supplier representatives should attend PI Planning to understand milestones and collaborate with dependent teams.

Suppliers should attend Solution Demos to present their accomplishments and provide feedback on the demos of the other trains.

Involvement in the I&A workshop is crucial since traditional suppliers will have longer learning cycles. They should use this opportunity to raise problems or issues and participate in the overall solution.

In addition, suppliers may have limited flexibility in adjusting their plans, and, as a result, other trains will have to adjust accordingly.

In addition, the solution's architecture may play a significant role in supplier collaboration. The article, [Technical Strategies for Agile and Waterfall Interoperability at Scale](#), discusses several practices that enable better collaboration with traditional suppliers.

Collaborating with Suppliers

Collaboration with suppliers occurs at all levels of SAFe, from formulating [Strategic Themes](#) to prioritizing Team backlogs. As described in [1], Honda collaborates with its suppliers on the kinds of products and markets it plans to pursue in the future. This increases alignment and enables their suppliers to offer better value through more strategic offerings. Instead of hiding information from suppliers, Honda shares it for better economic outcomes.

Suppliers also collaborate in formulating requirements, avoiding over-specification that can inhibit decentralized decisions and handcuff innovation. Instead of providing suppliers with detailed specifications, forward-thinking enterprises include them in the specification process. The supplier's teams should be treated as an extension of the ART and Solution Train teams.

The supplier's solution builders communicate directly with their customer counterparts along the aspects shown in Figure 2:

[Product Management](#) ensures the supplier's [Backlogs](#) and [Roadmap](#) align with the larger solution.

[System Architects](#) ensure technical alignment and a compatible solution design.

The [System Teams](#) share, build, integrate, and test infrastructure, including scripts, environments, and hardware. To enable early and frequent integration and improve quality, suppliers and ARTs need to share interfaces, tests, simulators, and other infrastructure.

[Agile Teams](#) frequently collaborate directly with their supplier counterparts to ensure alignment and remove bottlenecks on either side.

Figure 2. The Customer-supplier relationship is multi-dimensional

Figure 2. The Customer-supplier relationship is multi-dimensional

Helping Suppliers Improve

Changing to a Lean-Agile way of working significantly impacts the organization's supply chain, often disrupting years of collaborative experiences and relationships between organizations. For some enterprises, changing their supplier relationships may be as critical to achieving [Business Agility](#) as their own transformation.

To help support the transition, some organizations host a "Partner Day" event to explain why the organization is making the change and offer insights into future collaboration. Partners are invited to an open discussion to share concerns and expectations.

Lean-Agile enterprises also encourage suppliers to learn Lean-Agile ways to benefit both companies. They may include them in training events, allowing them to hear the same information and gain alignment with the internal teams. Knowledge transfer and growing T-shaped skills across the supply chain reduce bottlenecks

and improve flow (see [SAFe Principle #6 – Make Value Flow Without Interruptions](#)). Enterprises encourage organizations to collaborate and build knowledge between suppliers and their solution builders. This helps develop technical and process knowledge, improving delivery times, reducing costs, and enhancing relationships.

Agile Contracts

An effective working relationship with suppliers requires mutual trust. Traditional contracts can lead to undesirable results and unintended consequences. For example, a cost-plus contract award might yield a competitive, low-cost award, but result in overall cost increases or quality challenges as changes occur.

Building innovative systems with high uncertainty requires a different type of relationship and contracts that support it. Contract terms should include language on the new way of working, encourage shared responsibility for solution delivery, and foster win-win relationships. The [Agile Contracts](#) article provides more information on contracting to exploit the inherent variability in complex system development and how to make faster, easier changes to the supplier’s work as new knowledge emerges.

Learn More

[1] Liker, Jeffrey, and Thomas Y. Choi. *Building Deep Supplier Relationships*. Harvard Business Review, December 2004.

[2] Aoki, Katsuki, and Thomas Taro Lennerfors. *New, Improved Keiretsu*. Harvard Business Review, September 2013.

[3] Deming, W. Edwards. *Out of the Crisis*. MIT Center for Advanced Educational Services, 1982.

Last Update: 7 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Planning to Plan

ÑThe BobsÕ
whiteboard title from
the movie *Office
Space*

Pre-Plan

Pre-Plan describes the activities that align and prepare ARTs within a Solution Train for PI planning.

Adjust Cookie Settings

ARTs on a Solution Train plan their work through a series of common, cadence-

based PI Planning events. Aggregating these outcomes results in a PI Plan for the Solution Train (see [Roadmap](#)). The ART planning events may be conducted as a single “Solution Train PI Planning” event, or ARTs may plan in a more distributed fashion. In either case, Solution Train PI Planning is a critical synchronization point across all the ARTs and suppliers. This article describes the Pre-planning activities needed to prepare ARTs on a Solution Train for PI Planning.

Details

PI Planning is perhaps SAFe’s most powerful event as it creates business and technology alignment for an [Agile Release Train](#) (ART) and its stakeholders for the next increment of value delivery. In the context of a Solution Train, alignment before [PI Planning](#) is critical because multiple ARTs and suppliers are involved in individual or joint PI Planning sessions. [Solution Train](#) leaders must ensure their ARTs enter PI Planning aligned on a shared vision, goals, and objectives for the upcoming [PI](#).

Figure 1 shows the SAFe practices that help Solution Trains create and maintain alignment during PI execution, as described in the [Coordinate and Deliver](#) article. Pre-planning describes Solution Train leaders’ activities to align ARTs on the upcoming Solution Train [Capabilities](#) and cross-ART dependencies they will need to address during their PI Planning events.

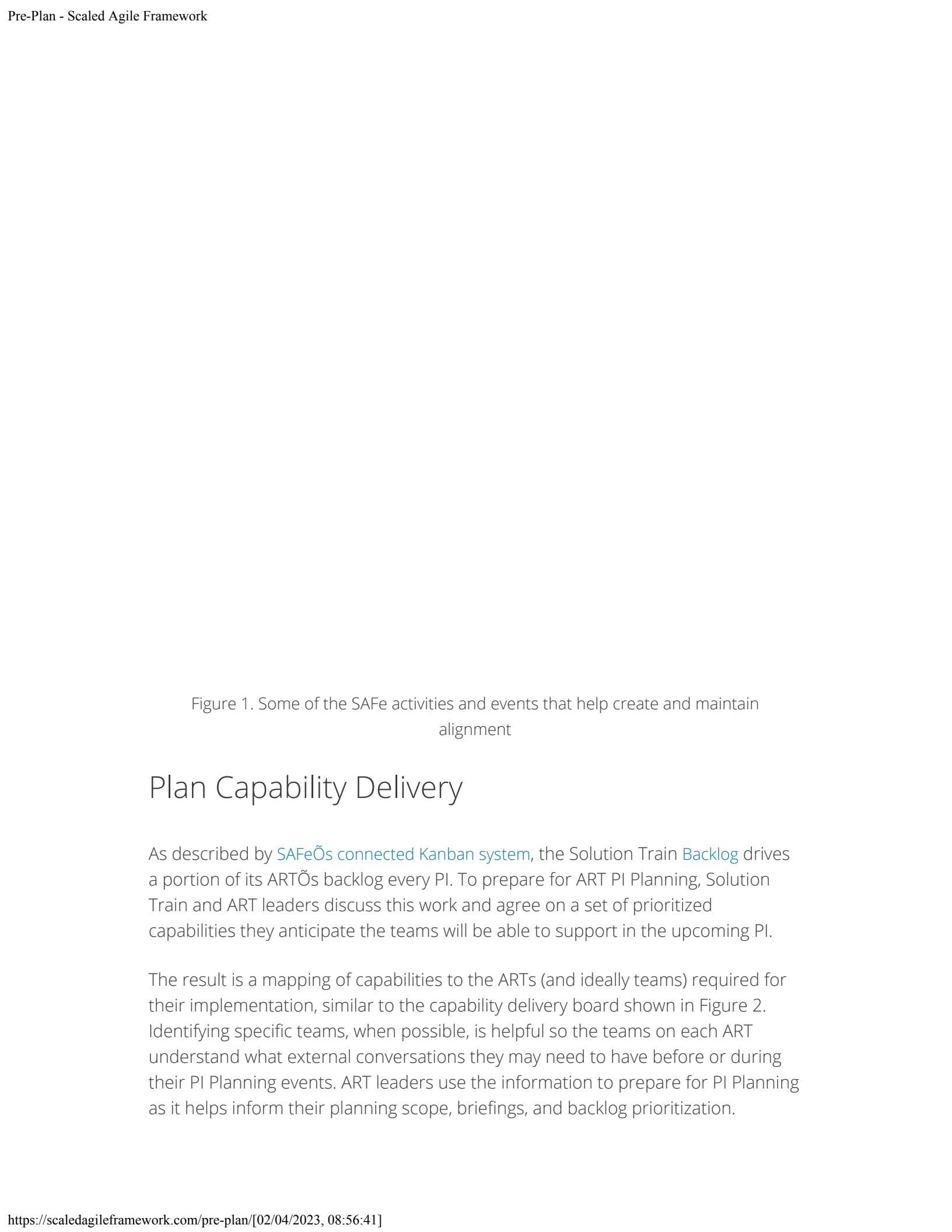


Figure 1. Some of the SAFe activities and events that help create and maintain alignment

Plan Capability Delivery

As described by [SAFe's connected Kanban system](#), the Solution Train [Backlog](#) drives a portion of its ART's backlog every PI. To prepare for ART PI Planning, Solution Train and ART leaders discuss this work and agree on a set of prioritized capabilities they anticipate the teams will be able to support in the upcoming PI.

The result is a mapping of capabilities to the ARTs (and ideally teams) required for their implementation, similar to the capability delivery board shown in Figure 2. Identifying specific teams, when possible, is helpful so the teams on each ART understand what external conversations they may need to have before or during their PI Planning events. ART leaders use the information to prepare for PI Planning as it helps inform their planning scope, briefings, and backlog prioritization.

Figure 2. Understand how ARTs and teams will deliver capabilities

Figure 2. The capability board shows how ARTs and teams will deliver capabilities

Understand and Map Dependencies

Aligning on capability delivery helps ART leaders identify some of the cross-ART dependencies for those capabilities. But ARTs have additional, local work driven by the [Solution Vision](#), [Solution Intent](#), and [Solution Roadmap](#). This local work may also have dependencies that Solution Train and ART leaders must identify before the ART PI Planning events.

A Solution Train dependency board shows work requested by one ART and provided by another in the upcoming PI. The board may also include internal and external suppliers. ARTs and their teams use this information to determine the scope and priorities for the upcoming work and to identify some of the cross-ART conversations required during their PI Planning events.

Figure 3. A Solution Train dependency board

Figure 3. A Solution Train dependency board

As it shows only dependencies, the dependency board does not replace the Solution Train planning board developed during PI Planning (see [Coordinate and Deliver](#)).

The Pre-Planning Workshop

A ÔPre-Planning WorkshopÕ is one way to create the alignment described above. Other SAFe events, such as the Solution Train Sync (see [Coordinate and Deliver](#)), also create alignment during the PI, which may be sufficient for the ARTs to enter PI Planning. The STE determines if a workshop is necessary based on the Solution TrainÕs and ARTÕs state of alignment. When a workshop is needed, most Solution Trains run them two to four weeks before PI Planning to provide the ARTs and teams with sufficient time to absorb the information and consider their coordination requirements for PI Planning preparation and execution.

Workshop attendees typically include the Solution Train and ART leaders, [Business Owners](#), and other relevant stakeholders. The workshop's goals and agenda vary widely depending on the solution under development, the Solution Train's current level of alignment, and the upcoming work. STEs use their facilitation skills to design and structure the workshop. They define the workshop's purpose, desired outcomes, and the activities to achieve them. Workshops often include breakout sessions for the different ART representatives to discuss specific topics. For example, the STE may use a breakout session to define the implementation strategy for an upcoming capability or address a dependency between ARTs and suppliers.

Either with or without a workshop, Pre-planning results in a shared understanding of the anticipated Solution Train level work and cross-ART dependencies for the upcoming PI. Solution Train and ART leaders create this information with sufficient time for the ARTs and teams to absorb it and consider their coordination requirements for PI Planning preparation and execution. See the [Coordinate and Deliver](#) article for more details on achieving alignment throughout the PI.

Last Update: 11 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit [Permissions FAQs](#) and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd, Suite 300
Boulder, CO 80301 USA

BUSINESS

[YouTube](#)

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*If everyone is
moving forward
together, success
takes care of itself.*

~Henry Ford

Coordinate and Deliver

Coordinate and Deliver describes the practices Solution Trains use to maintain the alignment and collaboration needed to continuously deliver value to large solution customers.

Building and operating effective large solutions requires the knowledge and expertise of many solution builders across diverse disciplines. Each ART and Supplier contributes unique knowledge and technical capabilities to deliver valuable customer solutions. Solution Train leaders and developers must continuously coordinate their activities to ensure consistent, predictable results.

Details

Alignment is not a natural state. Any lack of alignment on what the [Solution](#) is intended to doÑor how it will be achievedÑ is very costly. When building really big systems, every day lost represents significant financial waste and negative business impact. The traditional approach to achieving better alignment was creating detailed specifications and a detailed schedule. Unfortunately, this approach doesnÕt solve the problem. Indeed, it has significant adverse side effects:

- Constraining innovation
- Limiting the ability to explore alternatives
- Preventing adjustment from feedback and learning

In summary, the traditional approach did not serve the purpose of adequately reducing overall risk. Building large solutions that require a high degree of innovation and contain significant uncertainty requires a different approach. Rather than one-time alignment up-front, system builders must constantly align about exactly what the system to supposed to do and how it is supposed to do it.

The first step in creating alignment is to apply a common cadence to the building and deployment of these solutions. While [Agile Release Trains](#) (ARTs) and [Suppliers](#)Õ understanding may naturally diverge based on changing facts, local knowledge, and context, common PI boundaries support frequent realignment, as seen in the left side of Figure 1. But deviation, even during the PI, can also lead to significant waste and increased risks. Therefore, in addition to these shared PI timeboxes, solution builders need frequent, almost continuous alignment, as the right side of Figure 1 illustrates.

Figure 1. Solution Trains require continuous alignment

Figure 1. Solution Trains require continuous alignment

Solution Trains apply two mechanisms to create and maintain alignment throughout the PI, as shown in Figure 2. Solution Train *artifacts* define and communicate intent and forecasted plans, enabling *Agile Teams*, ARTs, and suppliers to make informed local decisions. Solution Train *activities* provide frequent alignment, assessment, and adjustment opportunities.



Figure 2. Solution Train artifacts and activities align ARTs and suppliers during PI execution

The remainder of this article describes these artifacts and activities.

Creating Alignment with Solution Train Artifacts

ARTs and teams require technical competence and organizational clarity to make informed, localized decisions (see [Principle #9 – Decentralize Decision Making](#)).

Solution Train artifacts provide clarity by defining and communicating intent and ensuring all solution builders have shared vision, goals, and objectives. The critical artifacts are:

Solution Vision ▷ defines the solution's purpose and how it solves customer needs

Solution Intent ▷ maintains and communicates the solution's requirements and design specifications

Solution Roadmap Ð delineates critical milestones and the solution's longer planning horizon

[Solution Management](#) and [Solution Architects](#) continually evolve these artifacts as ARTs and suppliers build the solution, and adjust them as new facts emerge. Everyone uses this information to drive their local backlogs and roadmaps while operating as fast and independently as feasible.

Creating Alignment with Solution Train Activities

The activities in Figure 1 can help maintain alignment during solution development. Each is described below.

Pre-Plan

Preparing for [PI Planning](#) is a significant alignment activity for a Solution Train. The [Pre-Plan](#) article describes how Solution Trains align and adjust objectives before the event based on the current state of the solution.

Plan the Solution Train PI

During PI Planning, Solution Trains must align their ARTs and suppliers on a joint plan for the upcoming PI. The individual ART planning events may be conducted as a single large event, or ARTs may plan in a more distributed fashion. In either case, Solution Train PI Planning is a critical synchronization point across all the ARTs and suppliers. Figure 3 illustrates a typical two-day pattern that can be used to create alignment during a Solution Train planning event.

Figure 3. A common Solution Train PI Planning pattern

Figure 3. A typical Solution Train PI Planning pattern

This format begins with Solution Train leaders presenting the Business Context to all teams across all ARTs. The ARTs continue their first day according to the ART PI planning agenda (performing product and architectural vision briefings, team breakouts, and so on). At the end of Day 1, representatives from each ART bring their “Management Review and Problem-Solving” outcomes to a Solution Train Management Review to coordinate their planning adjustments. On Day 2, the ART leaders present the aligned adjustments and continue planning following the Day 2 agenda. Consolidated “Readouts” occur across the entire Solution Train at the end of Day 2, so every Solution Train member sees the final plan.

But that is just one such pattern; Solution Train PI Planning can vary based on the dependencies between ARTs, planning sequences, and logistic concerns. For example, if the ARTs need to plan on different days or weeks, business context and the final plan readouts may need to be recorded and shared.

During planning, teams build a Solution Train planning board (Figure 4). This board is like the ART planning board (see [PI Planning](#)) but focuses on the [Capabilities](#) to be

delivered.

Figure 4. An example Solution Train planning board

Figure 4. An example Solution Train planning board

ARTs and Suppliers in a Solution Train depend on each other to deliver various solution elements. AS each dependency between ARTs is ultimately a dependency between teams, some Solution Trains find value in a more detailed planning board that shows team dependencies (Figure 5).

Figure 5. Solution Train planning board with team details

Figure 5. Solution Train planning board with team dependencies identified

When ART planning is complete, the [Solution Train Engineer](#) (STE) works with Solution Train and ART leaders to aggregate the ART's PI Objectives into Solution Train [PI Objectives](#). That creates the Solution Train PI Plan (see [Roadmap](#)), which defines the capabilities planned to be delivered in the upcoming PI.

Integrate Frequently

Agile development requires fast feedback. In turn, this requires frequent integration. However, the size, complexity, dependence on external suppliers, significant compliance and governance rules, and more make frequent integration of large solutions far more difficult.

In addition, the move to [DevOps](#) often suffers from legacy technologies, tightly coupled components, and a mixture of manual and automated build and test processes. In addition, cyber-physical systems often require hardware and materials that add long lead times to the integration mix.

Despite these potential impediments, frequent integration is necessary for the feedback and adjustment needed to reduce risk. While models, simulations, and lower-fidelity prototypes provide a fast and economical way to gain insights early in the product lifecycle (see [Model-Based Systems Engineering \(MBSE\)](#)), physical integration is still required. This requires a long-term strategy, investment in solution architecture, technology modernization, and the creation of an automated, end-to-end, [Continuous Delivery Pipeline](#).

Further, the cognitive load and infrastructure required to integrate these systems is often too large for the individual development teams. [System Teams](#) at both the ART and Solution Train levels help by building and supporting most of the [Continuous Integration](#) (CI) and [Continuous Deployment](#) (CD) infrastructure and automation (Figure 6). They enable fast feedback on changes by providing the following:

- Automation to build, integrate, and test the system at all levels
- Adequate environments for teams to test their part of the system
- Telemetry across the pipeline to measure performance and detect issues

Figure 7. Frequently integrate end-to-end changes

Figure 6. Frequently integrate the end-to-end solution with the CI/CD pipeline

Synchronize Routinely

Solution Trains are ÔnestedÕ teams of teams. High-performing teams create high-performing ARTs, which make high-performing Solution Trains. Scaling adds technical and social complexities that can block the teamsÕ work. ARTs and teams need mechanisms to communicate effectively and raise risks and impediments quickly to support continuous alignment. While any team can communicate with any other team at any time, scheduled synchronization events provide the frequent alignment needed to minimize divergence (Figure 7).

Figure 6. Teams and ARTs require a channel to raise issues for fast feedback

Figure 7. Teams and ARTs use cadence-based communication events to coordinate development and delivery

Like ARTs, Solution Trains have synchronization events that provide regular communication and coordination. Their purpose is to address issues and have the conversations necessary to keep the work moving forward and help teams resolve issues. These events include:

RTE Sync Ø similar to the ÔCoaches SyncÕ (See [PI article](#)), this event usually runs on the same cadence as the Coaches Syncs and is held soon after it to address issues that canÕt be resolved directly by the ARTs. Attendees include the STE, the [Release Train Engineers](#) (RTEs), select [Scrum Master/ Team Coaches](#), and others dictated by the agenda.

Product Manager Sync Ø similar to the PO Sync (See [PO Sync in PI article](#)), the Product Manager sync typically runs on the same cadence as the PO Sync and is held soon after it to raise and address issues from the ARTs. Attendees include [Solution Managers](#), [Product Managers](#), select [Product Owners](#), and others dictated by the agenda.

Architect Sync Ø a Solution Train event used to guide emerging designs, discuss tradeoffs, and increase opportunities to align implementation approaches without becoming a source of delay. Attendees include [Solution Architects](#), [System Architects](#), and team members. [Enterprise Architects](#) may also attend.

Participants in the sync events may vary. Architects often attend the PM Sync to provide input or learn from the discussions. Likewise, PMs may also participate in the Architect Sync. Each event is usually held weekly, with the ART events informing the agendas. The purpose is to manage change, resolve impediments, and make decisions about execution. Like the ART Sync, the RTE, PM, and Architecture Sync events may be combined into a Solution Train Sync when the topics overlap.

Demonstrate the Full Solution

Instead of measuring progress through stage gate milestones, schedules, and other proxies for value delivery, leaders provide input and feedback directly while participating in development events. [Solution Demos](#) are one such significant event where stakeholders observe and objectively evaluate the progress of the working, large solution. Solution Management often delivers these demos to provide the proper messaging and explanation of new capabilities.

Coordinate Releases

In an ideal environment, ARTs and Agile teams independently deploy and release small increments of value. However, large solutions have several challenges that inhibit frequent deployment and release:

Functional dependencies between features across ARTs and suppliers may require coordination

Long lead times for cyber-physical system components and waterfall-based suppliers may require more extensive release coordination

Marketing events may govern release dates and encourage a large-batch release of new functionality to spark publicity for these high-investment solutions

The Solution [Roadmap](#) and Solution Train planning board help maintain alignment by clearly communicating release dates and dependencies. To track and coordinate dependent work across the Solution Train, STEs and RTEs work with Solution and Product Management to help ARTs release large solution value. See “[releasing and release governance](#)” in the [Solution Train](#) article for more details.

Inspect & Adapt

Relentless improvement is a SAFe [Core Value](#). That commitment requires time and space for the people doing the work to improve their processes. Solution Trains benefit from their ART’s continuous improvement practices, including team retrospectives and [Inspect and Adapt](#) (I&A) events. However, some of the impediments discovered may require attention at the Solution Train level.

Solution Trains are too large for all team members across all ARTs to participate in improvement events like I&A, and not all members can contribute to all improvement items. For this reason, select individuals across the ARTs contribute to the Solution Train I&A based on the topic. The right people must attend through self-selection or invitation to get a good outcome.

Last Updated: 13 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit [Permissions FAQs](#) and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Linkedin](#)

[YouTube](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework

 Trust, but verify.

ÑRonald Reagan,
citing a Russian
proverb

Compliance

Compliance refers to the strategy, activities, and artifacts that allow teams to apply Lean-Agile development methods to build systems that have the highest possible quality, while simultaneously ensuring they meet regulatory, industry, and other relevant standards.

Adjust Cookie Settings

Enterprises use SAFe to build some of the world's largest and most important

systems, the failure of which may have unacceptable social or economic costs. These high-assurance systems include medical devices, automobiles, avionics, banking and other financial services, aerospace, and defense. To protect public safety, these systems are often subject to extensive regulatory or customer oversight and rigorous compliance requirements. In addition, many enterprises are subject to government regulations (examples: Sarbanes-Oxley, HIPAA, ACA, state insurance regulations) that require similar attention and auditing to ensure compliance.

Organizations operating under such regulations have relied on comprehensive quality management systems (QMS). However, legacy QMS systems are based on traditional, phase-gated development models that cannot keep pace with accelerating time-to-market demands. Of more significant concern is that even when the higher Cost of Delay (CoD) is accepted, these traditional approaches often do not increase quality or eliminate risk. As Deming notes, “Inspection is too late. The quality, good or bad, is already in the product.”

One of the ten practices of SAFe’s [Enterprise Solution Delivery](#) competency is “continually addressing compliance concerns.” This article extends that knowledge and offers guidance on applying Lean-Agile methods to build these systems faster and better while addressing critical compliance requirements.

Note: A white paper supports this article, “Achieving Regulatory and Industry Standards Compliance with the Scaled Agile Framework™ (SAFe®),” which you can [download here](#). In addition, there is a SAFe webinar on the topic, with Q&A, which can be [viewed here](#).

Details

Traditional waterfall practices often mandate that full system specifications are defined and committed to in detail, up-front, long before all the actual needed system behaviors can be known. Worse, the sequential nature of phase-gate development produces large batches of work, long cycles between system integration points, and late feedback. In addition, compliance activities are typically deferred until the end of the project, providing little insight into compliance progress.

This often results in missed deadlines, disappointing business or mission outcomes, lower quality, and substantial (and late) compliance challenges. In contrast, high-assurance Lean-Agile development builds in quality incrementally—early and throughout the development lifecycle.

The Role of the Quality Management System

To satisfy compliance requirements, organizations must demonstrate that their system meets its intended purpose and has no unintended consequences that might cause harm. They must also develop the objective evidence required to prove that the system conforms to those standards. To that end, organizations that build high-assurance systems define their approved practices, policies, and procedures in a QMS. These systems ensure that development activities and outcomes comply with all relevant regulations and provide the required documentation to prove it.

Unfortunately, many QMS systems are heavily influenced by traditional phase-gated waterfall methods. This seriously inhibits, and can even prevent, the adoption of newer techniques, as the older methods are hard-coded into the *only* approved way of working. As Figure 1 illustrates, SAFe describes an incremental approach to development and compliance that includes five recommended practices.

Figure 1. A Lean-Agile quality management system improves quality and makes compliance more predictable

Clearly, those who want the benefits of Lean-Agile development (faster time to market and higher quality, to name a few) will have to evolve a Lean QMS.

The remainder of this article provides guidance on these five specific practices teams can use to achieve high-assurance systems compliance.

Build the Solution and Compliance Incrementally

Even with a set of robust specifications, [Agile Teams](#) never have all the answers when development begins. Instead, they have a set of hypotheses that must be tested through a series of short, iterative experiments, providing validated learning to advance toward the ultimate solution. Figure 2 highlights SAFe's incremental development approach, comparing Shewhart/Deming's Plan-Do-Check-Adjust (PDCA) learning cycles with a traditional waterfall model.



Figure 2. Rapid Plan-Do-Check-Adjust learning cycles increase system quality and reduce compliance risk

Figure 2 illustrates two important implications for compliance. First, building smaller, working parts of the solution early allows compliance activities to also begin early, removing the large set of such actions at the end. Each increment assesses both the viability of the current solution and its progress toward compliance, providing early feedback on the system's ultimate fitness for use. Second, specifications are created early and evolve over time in small batches, with faster feedback on decisions and the opportunity for continuous review and assessment.

Organize for Value and Compliance

Agile Release Trains (ARTs) are the primary value delivery organizations in SAFe. Each train requires all the skills necessary to build and release the Solution, including those responsible for Quality Assurance (QA), security, testing, and

Verification and Validation (V&V). (Note: While some regulations require independent, objective assurance, compliance representatives can still participate continuously as ART members). The result is an ART that includes compliance competency, as illustrated in Figure 3.

Figure 3. Agile Release Trains include all disciplines, including compliance

Figure 3. Agile Release Trains include all disciplines, including compliance

Solution and Product Management ensure that the [Solution Intent](#) and backlog properly reflect compliance requirements. Teams also ensure that their work includes appropriate compliance activities.

Build In Quality and Compliance

Built-In Quality is a dimension of SAFe's [Team and Technical Agility](#) competency and a core tenet of the Lean-Agile Mindset. SAFe describes using [Built-In Quality](#) practices, including automation, to detect compliance and quality problems. This

philosophy applies [Systems Thinking](#) by Œoptimizing the whole,Œ ensuring fast flow across the entire [Development Value Stream](#), and making quality everyoneŒs job. Quality becomes a culture, not a job title.

To that end, compliance concerns are also built directly into the development process, and automated wherever possible, as illustrated in Figure 4.

Figure 4. Automating compliance with design-build-test automation

Figure 4. Automating compliance with design-build-test automation

However, not all compliance activities can be automated, as some regulatory requirements mandate manual activities, such as Failure Mode and Effects Analysis (FMEA) and audits. This work is planned as part of the [Team Backlog](#) as part of the regular flow of work. The goal is to conduct these activities and reviews as the solution is being built, reducing the last sign-off activity from a significant, extended event to a quick and boring Œnon-event.Œ

This approach provides early feedback on the degree to which the teamŒs

compliance activities are being performed and how those activities may impact team performance. Figure 5 shows the feedback cycle between compliance and the practices defined by the Lean QMS.

Figure 5. PIs and retrospectives provide feedback loops for compliance activities

Continuously Verify and Validate (V&V)

High-assurance systems require Verification and Validation (V&V) to ensure that:

- The system works as designed (verification)
- It meets the intent of its purpose and functions (validation)

V&V must always occur against a known set of requirements. Otherwise, there is nothing to V&V. As Figure 6 illustrates, SAFe uses solution intent as the repository for existing and emerging requirements and designs that are the subject of V&V.

Figure 6. SAFe®s Solution Intent provides support for Verification and Validation

Traceability within the solution intent ensures that the artifacts produced—the actual software, hardware components, etc.—always address regulatory and compliance specifications, providing end-to-end evidence that V&V requirements have been met.

The SAFe Requirements Model Supports V&V

To assure compliance, all material requirements elements (feature, story, NFRs) have test cases that are created at the same time as the functionality (Figure 7).



Figure 7. SAFe's requirements meta-model supports Verification and Validation

Each increment yields new functionality and, consequently, adds new tests. As the number of tests grows, automation is vital to prevent testing activities from becoming bottlenecks.

Make V&V and Compliance Activities Part of Flow

To improve [flow](#), V&V validation activities are performed in small batches. Each work item includes all the necessary information required for compliance. Any reviews, audits, and sign-offs are included in the work item's definition of done (DoD), as shown in Figure 8. Each PI, the System Demo evaluates the solution's progress toward its operational goals *and* the objective evidence necessary to assure compliance.



Figure 8. Verification and Validation activities are integral to a continuous flow of value

Release Validated Solutions on Demand

Finally, although the product development process happens on a predictable cadence (See [Principle #7- Apply cadence, synchronize with cross-domain planning](#)), the release process may require additional activities. These can include:

- Validation testing of the final release candidate (examples: medical trial, flight test)
- Review of the objective evidence needed before production approval and release
- Customer and user acceptance tests, document submissions, regulatory approvals

These can be extensive activities and must be included in planning and development. Even then, Lean-thinking organizations constantly strive to fully automate the delivery and, wherever possible, build in automated final release checks as part of a SAFe [Continuous Delivery Pipeline](#) and [Release on Demand](#).

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements*. Pearson Education, 2011.

Last update: 30 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework

Achieving Regulatory and Industry Standards

Compliance with SAFe

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

[Adjust Cookie Settings](#)

Achieving Compliance with Lean-Agile Principles

SAFe compliance is shown as part of [Solution Intent](#) at the [Large Solution Level](#) on the Big Picture (see Figure 1.) Clicking on the [Compliance](#) text will link you to the standard guidance in SAFe. It describes the strategy, activities, and artifacts that allow teams to apply Lean-Agile development methods to build systems with the highest possible quality while ensuring they meet regulatory, industry, and other relevant standards.

Figure 1. Solution intent supports compliance.

A detailed white paper, *Achieving Regulatory and Industry Standards Compliance with the Scaled Agile Framework™ (SAFe™)*, supplements this guidance, which you can [download here](#). There is also a SAFe webinar on the topic, with Q&A, which can be watched [here](#).

SAFe Regulatory and Compliance Toolkit

The SAFe Regulatory and Compliance Toolkit is for SAFe Practice Consultants (SPCs) who must present SAFe in the context of industries and programs that must demonstrate compliance. It includes a PPT presentation as the primary tool, referencing the white paper and webinar. SPCs can download the compliance toolkit on the SAFe Community Platform [here](#).

Last Updated: 26 February 2023

images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



You just have to have the guidance to lead you in the direction until you can do it yourself. It is the neglect of timely repair that makes rebuilding necessary.

~Richard Whately

Refactoring

Adjust Cookie Settings

Note: This article is part of [Extended SAFe Guidance](#) and represents official

SAFe content that cannot be accessed directly from the Big Picture.

Refactoring is an activity to improve a code or component's internal structure or operation without changing its external behavior. The goal of software development is the continuous delivery of business value to users and stakeholders. Constantly changing technology and evolving business objectives make maintaining and continuously increasing business value difficult. Two paths to the future exist:

1. Keep adding new functionality to an existing code base toward an eventually unmaintainable "throw-away" state.
2. Continuously modify the system to provide a foundation for efficiently delivering *current* and future business value.

The second choice, refactoring, is better. With continuous refactoring, the useful life of an *Enterprise*'s investment in software assets can be extended as long as possible. Users can continue to experience a flow of value for years to come. Refactors enable an *emergent design*, ensuring the system continues to meet future business needs. Refactors are a particular type of *Enabler* story in SAFe, and, like any other *Story*, they must be estimable, verifiable, and valuable, as well as accepted by the *Product Owner*.

Details

Figure 1 illustrates the fundamentals of *refactoring*, which is modifying any software entity—a module, method, or program—to improve its structure or viability without changing its external functionality.

Figure 1. Refactoring in an isolated environment for change within a larger entity

For example, refactors may accomplish such things as increases in processing speed, sourcing different internal data, or improving security concerns. Another type of refactoring involves streamlining some aspects of the code to make it more efficient, maintainable, or readable.

Refactoring requires that each change is tested immediately to verify the accomplishment of the desired goal. A refactor may be broken into sequential micro-refactors to accomplish a more important purpose. Each small refactor must be tested to ensure correctness. This iterative process preserves the integrity of the software at any stage.

SAFe emphasizes the importance of keeping all work visible, including refactoring. Like user value work, refactoring must be planned for, estimated, and prioritized.

Sources of Refactors

Refactors arise from various sources, as illustrated in Figure 2.

Figure 2. Possible sources of refactors

Figure 2. Possible sources of refactors

A refactor can be provoked by a business [Feature](#) or a part of a larger refactoring initiative required by some new architectural enabler. New user stories may also require some refactoring of code. Technical debt may drive the team to refactor specific components. Some may be necessitated by new [Nonfunctional Requirements](#) (NFRs)

Not all refactoring efforts originate from a story. The [Test-Driven Development \(TDD\)](#) process encourages continuous refactoring as part of making code changes. Developers are constantly updating their designs to support current and upcoming requirements better. Such work should be factored into the estimate of the corresponding [Story](#) (see continuous refactoring techniques described in [1], [2], [3]). Specific refactors, however, represent more significant pieces of a redesign that needs to be planned and tracked as separate backlog items.

Specifying Refactors

Understanding what value will be achieved once the refactoring is completed is essential. Teams may wish to apply the user story voice form to foster a shared understanding of purpose and value, as Figure 3 illustrates:

Figure 3. Refactor example

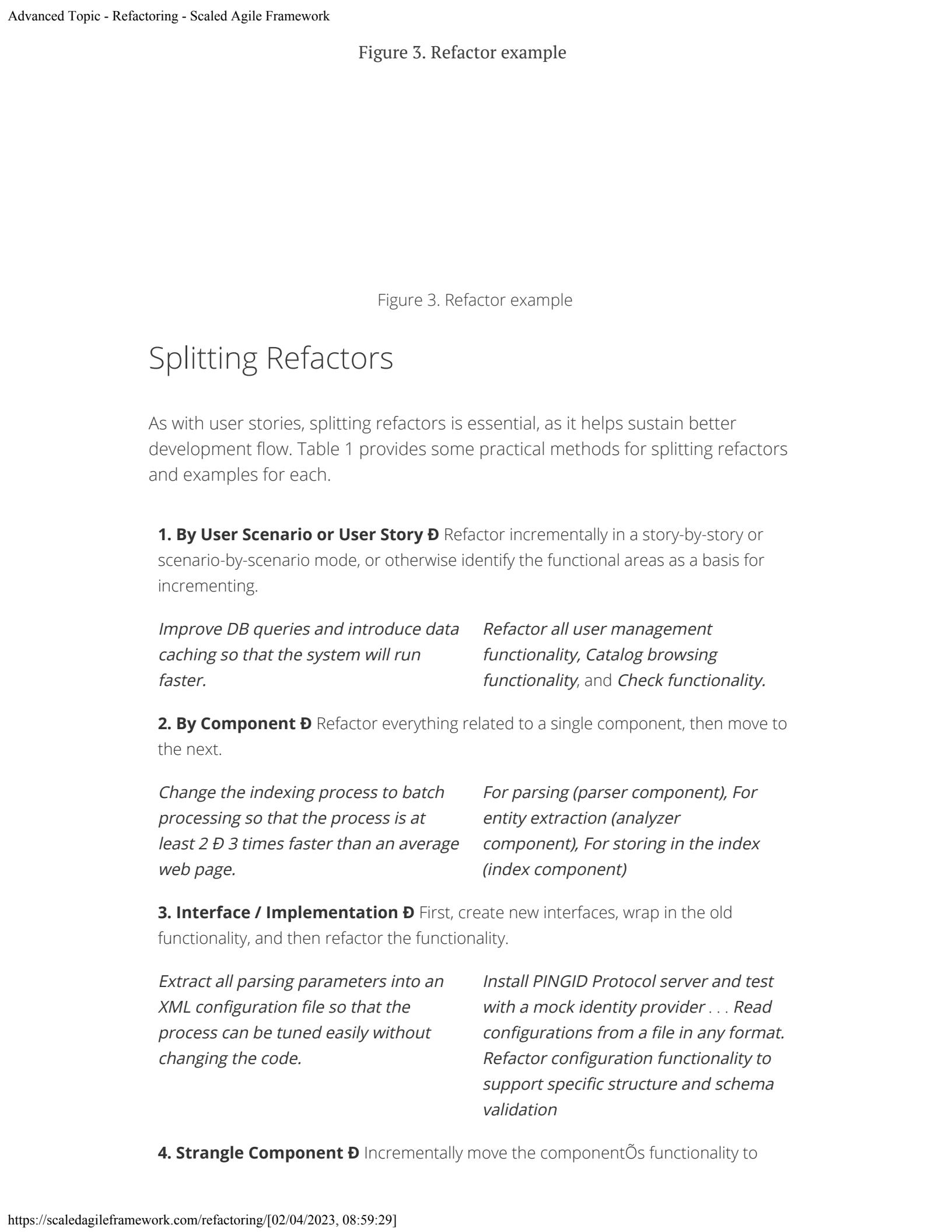


Figure 3. Refactor example

Splitting Refactors

As with user stories, splitting refactors is essential, as it helps sustain better development flow. Table 1 provides some practical methods for splitting refactors and examples for each.

1. By User Scenario or User Story Refactor incrementally in a story-by-story or scenario-by-scenario mode, or otherwise identify the functional areas as a basis for incrementing.

<i>Improve DB queries and introduce data caching so that the system will run faster.</i>	<i>Refactor all user management functionality, Catalog browsing functionality, and Check functionality.</i>
--	---

2. By Component Refactor everything related to a single component, then move to the next.

<i>Change the indexing process to batch processing so that the process is at least 2 ÷ 3 times faster than an average web page.</i>	<i>For parsing (parser component), For entity extraction (analyzer component), For storing in the index (index component)</i>
---	---

3. Interface / Implementation First, create new interfaces, wrap in the old functionality, and then refactor the functionality.

<i>Extract all parsing parameters into an XML configuration file so that the process can be tuned easily without changing the code.</i>	<i>Install PINGID Protocol server and test with a mock identity provider . . . Read configurations from a file in any format. Refactor configuration functionality to support specific structure and schema validation</i>
---	--

4. Strangle Component Incrementally move the component's functionality to

other components; once everything is moved, delete the old code.

Replace the database with a custom search index so indexing and search performance improve 10 ÷ 20 times.

Move index data to the custom search index first. Move entity dictionaries.

5. Inline Refactoring / Extraction Refactor the functionality inline where it currently is, but then extract it and encapsulate it in a component, class, or method/function.

Replace current ad hoc parsing with grammar-based functionality so that changing parsing rules would become easy without coding.

Refactor the code (as is) to use grammar notation. Extract all grammar-related functionality into a grammar engine.

Table 1. Methods of splitting refactor stories

Establishing acceptance criteria

As with user stories, defining acceptance criteria for refactors helps resolve ambiguities. Figure 4 illustrates the additional specificity that comes with established acceptance criteria.

Figure 4. Example of acceptance criteria for a refactoring story

Figure 4. Example of acceptance criteria for a refactoring story

Acceptance criteria can often be used as a basis for splitting. For example, the first step in Figure 4 might be to make synchronous non-configurable batch processing with a single query to the dictionary but without the debug logging. Then add the capability to read the batch size from the file. Step 3 would be the process items asynchronously, and finally, add debug logging functionality.

Demonstrating Refactors

Even though refactoring is focused on the internal working of the code, as with any other story, teams demonstrate the results. From the example above, the teams might show the following:

1. Reduced processing time on a few web pages compared to the previous benchmark
2. Dependency of processing time on the size of the batch, which can be configured from the file

3. A code snippet for asynchronous processing
4. The debug log file that captures all the operations
5. The number of queries to dictionaries per batch (from the log file)

Adopting a Refactoring Culture

Refactoring is a needed skill for [Agile Teams](#) and is a critical component of the [Team and Technical Agility](#) competency of [Business Agility](#). Refactors should routinely appear on the [Team Backlog](#) and be included—along with inline refactoring—in story estimates. A Design [Community of Practice](#) (CoP) can foster awareness and attention to refactoring techniques. [Scrum Master/Team Coaches](#) can help their teams learn effective approaches to specifying, estimating, and splitting refactors. [Product Owners](#) should embrace refactoring by prioritizing the work and helping define acceptance criteria.

A test automation culture, including TDD and [Behavior-Driven Development](#) (BDD), creates a large set of tests that make refactoring more accessible and reliable. Errors introduced by refactoring are caught immediately by the tests.

Learn More

[1] Fowler, Martin, et al. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999.

[2] Martin, Robert. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008.

[3] Wake, William. *Refactoring Workbook*. Addison-Wesley, 2003.

Last update: 24 February 2023

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



*It's just what I asked
for, but not what I want.*

~*The Night Before
Implementation* poem,
Author Unknown

Behavior-Driven Development

Note: This article is part of [Extended SAFe Guidance](#) and represents official SAFe content that cannot be accessed directly from the Big Picture.

[Adjust Cookie Settings](#)

Behavior-Driven Development (BDD) is a test-first, [Agile Testing](#) practice that provides [Built-In Quality](#) by defining (and potentially automating) tests before or as part of specifying system behavior. BDD is a collaborative process that creates a shared understanding of requirements between the business and the [Agile Teams](#). Its goal is to help guide development, decrease rework, and increase flow. Without focusing on internal implementation, BDD tests are business-facing scenarios that attempt to describe the behavior of a Story, Feature, or Capability from a user's perspective.

When automated, these tests ensure that the system continuously meets the specified behavior *even as the system evolves*. That, in turn, enables [Release on Demand](#). Automated BDD tests can also serve as the definitive statement regarding the as-built system behavior, replacing other types of behavioral specifications.

Details

Align on System Behavior

Aligning on *precisely* what to build is a challenge when developing innovative systems. In addition, new ideas are difficult to communicate with the diverse stakeholders responsible for system implementation. Figure 1 illustrates the three perspectives (called the *triad* [1]) required to define solution behavior clearly:

Customer-centric stakeholders understand customer and business needs and the relative desirability and viability of a new requirement

Development-centric stakeholders understand the solution space and technological feasibility

Test-centric stakeholders consider the exceptions, edge cases, and boundary conditions for the new behavior

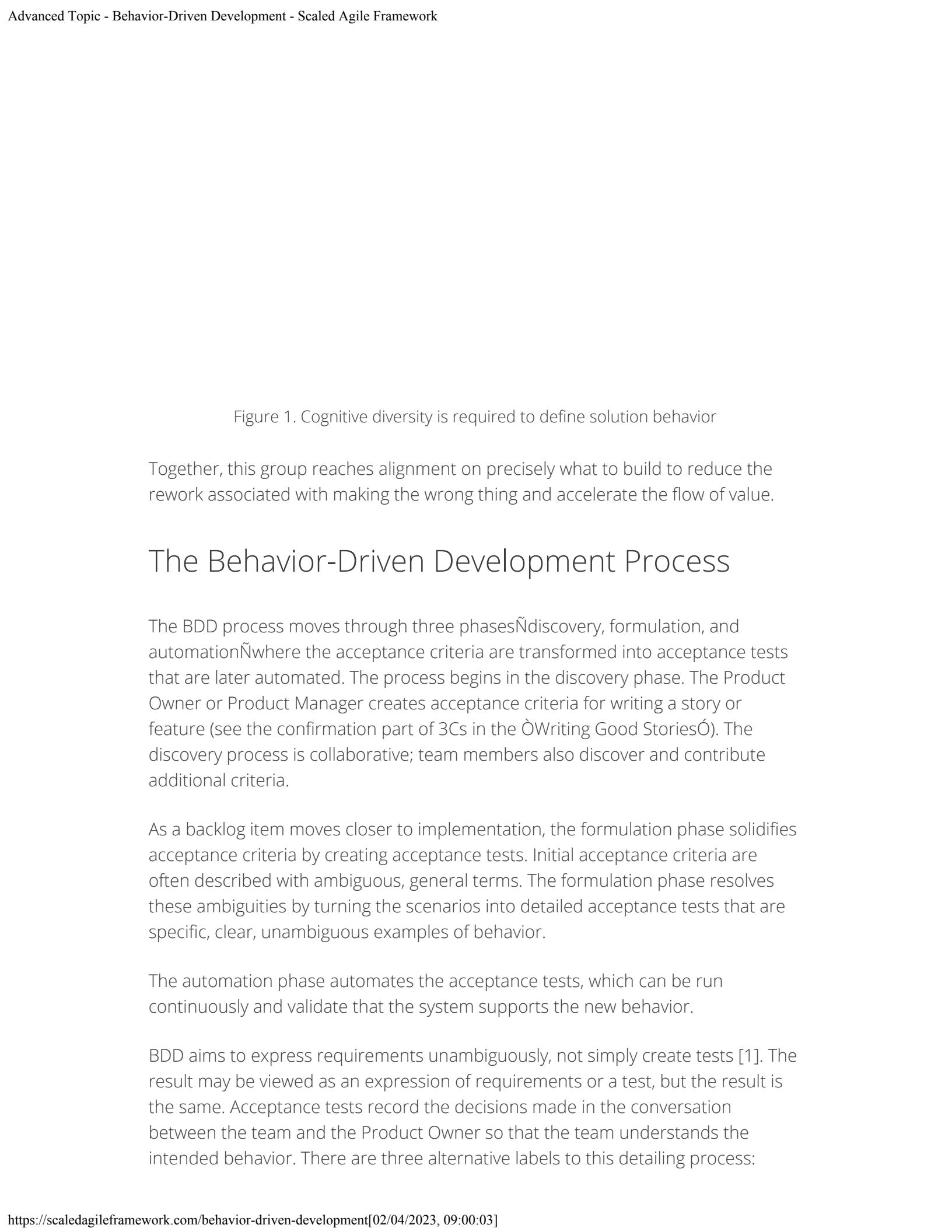


Figure 1. Cognitive diversity is required to define solution behavior

Together, this group reaches alignment on precisely what to build to reduce the rework associated with making the wrong thing and accelerate the flow of value.

The Behavior-Driven Development Process

The BDD process moves through three phases—discovery, formulation, and automation—where the acceptance criteria are transformed into acceptance tests that are later automated. The process begins in the discovery phase. The Product Owner or Product Manager creates acceptance criteria for writing a story or feature (see the confirmation part of 3Cs in the “Writing Good Stories”). The discovery process is collaborative; team members also discover and contribute additional criteria.

As a backlog item moves closer to implementation, the formulation phase solidifies acceptance criteria by creating acceptance tests. Initial acceptance criteria are often described with ambiguous, general terms. The formulation phase resolves these ambiguities by turning the scenarios into detailed acceptance tests that are specific, clear, unambiguous examples of behavior.

The automation phase automates the acceptance tests, which can be run continuously and validate that the system supports the new behavior.

BDD aims to express requirements unambiguously, not simply create tests [1]. The result may be viewed as an expression of requirements or a test, but the result is the same. Acceptance tests record the decisions made in the conversation between the team and the Product Owner so that the team understands the intended behavior. There are three alternative labels to this detailing process:

Behavior Driven Design (BDD)
Acceptance Test-Driven Development (ATDD),
Specification by Example (SBE)

Although slight differences exist in these approaches, they all emphasize understanding requirements before implementation.

A Behavior-Driven Development Example

Behavior description begins with a story, feature, or capability specified by its acceptance criteria. All of these are defined using terms from the customer's domain, not from the implementation. Here is an example story and its acceptance criteria:

Figure 2. An example Story and acceptance criteria

Figure 2. An example Story and acceptance criteria

The acceptance criteria could also be written in *Given-When-Then* (GWT) format as shown below:

Given a speed limit

When the car drives

Then it is close to the speed limit but not above it

Even then, elaborated acceptance criteria are typically insufficient to code the story. To remove ambiguity, formulate the scenario into one or more examples that specify the details of the behavior, resulting in a specific acceptance test:

Given speed limit is 50 mph

When the car drives

Then its speed is between 49 and 50 mph

In collaboration with the team (the triad), additional acceptance criteria and scenarios will emerge; for example: *When the speed limit changes, the speed changes without excessive force.*

This criterion results in an additional test (or tests) that stipulate what excessive deceleration is acceptable:

Given speed limit is 50 mph

When the speed limit changes to 30 mph

Then deceleration rate should be less than 5 feet/sec/sec

Figure 3 illustrates the BDD process that begins with a story and details its specification in two dimensions. Horizontally, additional acceptance criteria describe the story's requirements. Vertically, other acceptance tests detail those acceptance test requirements.

Figure 3. The BDD process details behavioral specifications

Figure 3. The BDD process details behavioral specifications

Automating Acceptance Tests

Automating these business-facing tests is an important reason to use the Given-When-Then format. Frameworks, including Cucumber and ÔFramework for Integrated TestingÕ (FIT), can be used to support this syntax. To support regression and continuous delivery, tests should be automated wherever possible.

Story acceptance tests are written and executed in the same iteration as the code development. If a story does not pass its tests, the team does not receive credit for that story. Features and capabilities have acceptance tests showing how several stories work together in a broader context. Typically, these tests represent the behavior of more significant workflow scenarios and should run during the **iteration** when the feature or capability is finished.

Learn More

[1] Pugh, Ken. *Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration*. Addison-Wesley, 2011.

Last update: 27 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

[CONTACT US](#)
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Cadence and synchronization limit the accumulation of variance.

—Don Reinertsen,
Principles of Product Development Flow

Principle #7 ⚡ Apply cadence,

Adjust Cookie Settings

synchronize with cross-domain planning

Solution development is an inherently uncertain process. If it weren't, the solutions would already exist, and there would be no room for the next generation of innovations. This uncertainty conflicts with the need for businesses to manage investments, track progress, and have sufficient confidence in future outcomes to plan and commit to a reasonable course of action.

To address this challenge, Agile development functions best in a "safety zone," where enough uncertainty provides the freedom to pursue innovation and react to events while providing the confidence the business needs to operate. The primary means to achieve this balance is through objective knowledge of the current state. It's just more manageable when you know where you really are. This knowledge is gained by applying *cadence*, *synchronization*, and periodic *cross-domain planning*.

Cadence is a rhythmic pattern of events that provides the steady heartbeat of the development process. It makes routine everything that can be routine, so developers can focus on managing the variable part of solution development.

Synchronization allows multiple solution perspectives to be understood, resolved, and integrated at the same time.

Figure 1 below highlights the benefits of cadence and synchronization.

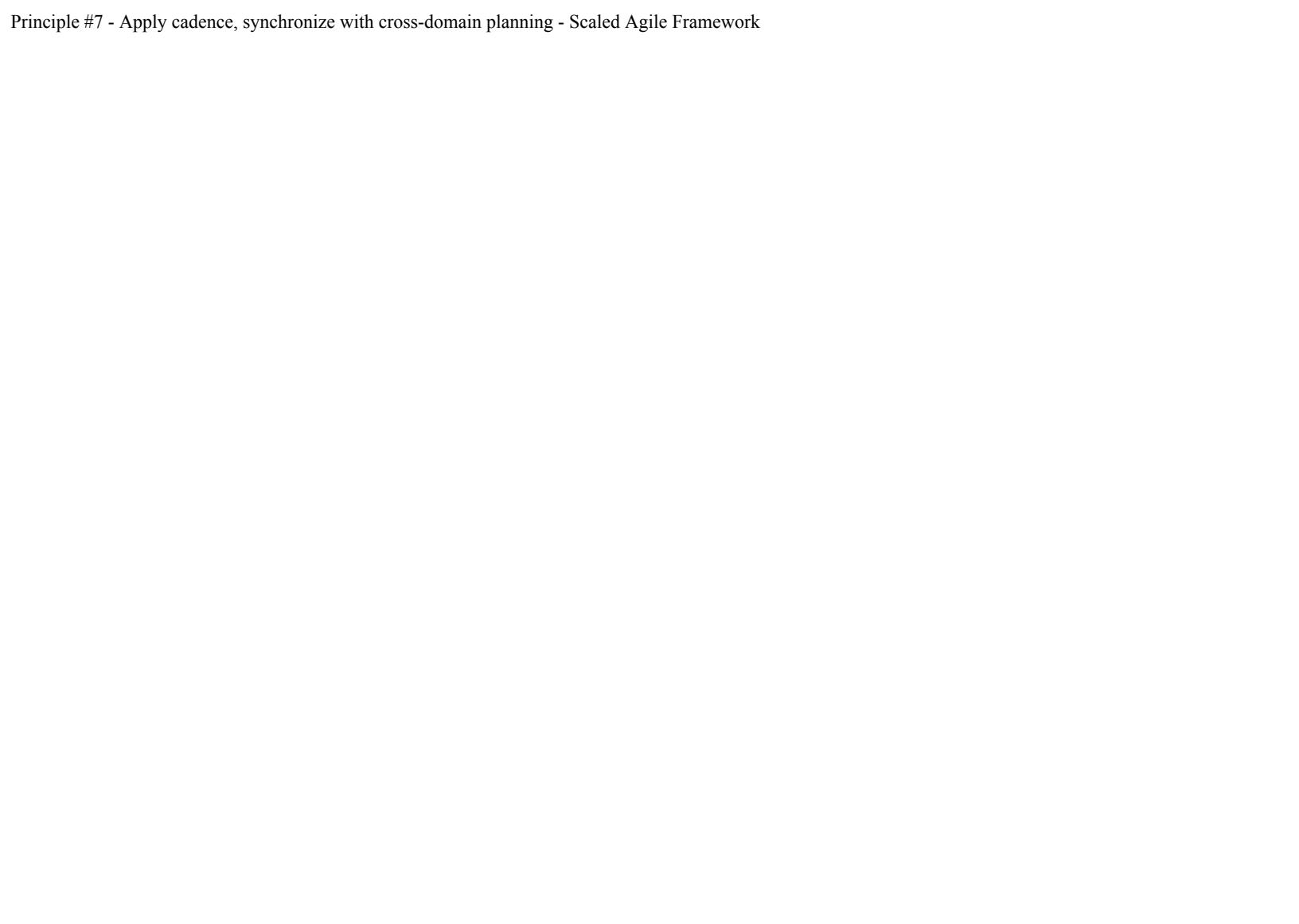


Figure 1. The benefits of development cadence and synchronization in development

The combination of cadence and synchronization helps Agile teams proceed confidently despite the inherent uncertainty.

Cadence and synchronization enable flow

Thanks to Don Reinertsen's *principles of product development flow* [1], we can explain why cadence and synchronization are critical to effective solution development. Tables 1 and 2 below summarize some of these principles and the relevant SAFe practices that implement them.

Principles of Flow: Cadence

F5: Use a regular cadence to limit the accumulation of variance

SAFe Practices

Planning at regular PI intervals limits variances to a single PI timebox, increasing Agile Release Train (ART) and

	Solution Train predictability.
F6: Provide sufficient capacity margin to enable cadence	To reliably meet PI objectives, the Innovation and Planning (IP) iteration has no planned scope and provides a schedule margin (buffer). In addition, planned-for but uncommitted objectives provide capacity margin (scope buffer). Together, they offer a way to meet PI goals reliably.
F7: Use cadence to make waiting times predictable	If a Feature doesn't make it into a PI but remains a high priority, its delivery can be scheduled for the following PI (or another scheduled, frequent release). This avoids the temptation to load excess Work in Process (WIP) into the current increment.
F8: Use a regular cadence to enable small batch sizes	Short iterations help control the number of Stories in the iteration batch. Feature batch sizes are controlled by short PIs and frequent releases, providing high system predictability and throughput.
F9: Schedule frequent events using a predictable cadence	PI planning, System Demos, Inspect and Adapt (I&A), ART Sync, Iteration Planning, backlog refinement, and architecture discussions benefit from frequent events. Each event needs to process only a small batch of new information. Cadence helps lower the transaction costs of these events.

Table 1: Cadence principles applied in SAFe

Principles of Flow: Synchronization	SAFe Practices
F10: Exploit economies of scale by synchronizing work from multiple projects	Individual Agile Teams are aligned to common iteration lengths. Work is synchronized by system and solution demos. Portfolio business and Enabler Epics drive common infrastructure and Customer utility.

F11: Capacity margin enables synchronization of deliverables	The Innovation and Planning (IP) iteration enables the final PI system demo and solution demo to occur without taking velocity away from ARTs or Solution Trains.
F12: Use synchronized events to facilitate cross-functional trade-offs	ART and Solution Train events synchronize customer feedback and enable resource and budget adjustments, mission alignment, continuous improvement, and program oversight and governance. They also drive collaboration and team building.
F13: To reduce queues, synchronize the batch size and timing of adjacent processes	Teams are aligned to common timeboxes and similar batch sizes. The ART and solution system teams support integration on a regular cadence. Backlogs are kept short and uncommitted to facilitate the rapid delivery of new ideas.
F14: Apply nested cadence harmonic multiples to synchronize work	Teams integrate and evaluate at iteration boundaries (at least). ARTs and Solution Trains evaluate at PI boundaries.

Table 2: Synchronization principles applied in SAFe

Cadence and synchronization are critical concepts that help us manage the inherent variability of our work. Together they create a more reliable, dependable solution development and delivery process that our key business stakeholders can rely on.

Aligning Development Cadence

Agile teams naturally apply cadence via short iterations (or Sprint) cycles. However, even when working together, these cadence-based cycles may not lend themselves to effective synchronization, as Figure 2 illustrates.

Figure 2. Agile teams operating on different cadences

Figure 2. Agile teams operating on different cadences

As Figure 2 illustrates, each team is “sprinting” on a cadence, but the cadences are different, and the system may or may not be evolving as planned. Operating on a common cadence—and adding the synchronization of routine system demos—addresses this concern, as illustrated in Figure 3

Figure 3. Common cadence supported by regular system demos

Figure 3. Common cadence supported by regular system demos

In Figure 3, the system is now truly Ôsprinting,Õ as is illustrated by the objective evidence from the regular system demo.

Synchronize with Cross-Domain Planning

In addition to common cadence, periodic cross-domain planning (for example, PI Planning in SAFe) provides the opportunity for the various aspects of a solution—business and technical—to be integrated and evaluated together at one time. The net effect is managing variability by frequently revisiting and updating the plan. In other words, cadence-based planning limits variability to a single interval, as Figure 4 illustrates.

Figure 4. Cadence-based planning limits the variability of the plan to actual

Figure 4. Cadence-based planning limits the variability of the plan to actual

With infrequent or one-time planning, there is no consistent mechanism to validate that the actual solution aligns with the plan. Deviations can accumulate over time to the point that the cost of correcting the variations can be exorbitant. With synchronized cross-domain planning, the reality of what has been built is compared to the plan more frequently and validated through demonstration of working system functionality. Variances are smaller, easier, and less costly to correct in the subsequent interval. In this way, the Ôplan is far closer to reality.Õ Business conditions are significantly improved when everyone operates according to a current and realistic plan, which is tuned and adjusted as new facts emerge.

Summary

ThereÕs no cure for the inherent uncertainty of solution development. If there were, it would surely be worse than the disease. However, cadence and synchronization, along with periodic cross-domain planning, provide the tools

needed to operate in the safety zone.

Learn More

[1] Reinertsen, Donald. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

[2] Kennedy, Michael. *Product Development for the Lean Enterprise*. Oaklea Press, 2003.

Last update: 22 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK	PARTNER	CONTENT & TRADEMARKS	SCALED AGILE, INC
Download SAFe Posters & Graphics	Becoming a Partner	FAQs on how to use SAFe content and trademarks	CONTACT US
Watch and download SAFe videos and presentations	Partner Directory	Permissions Form	5400 Airport Blvd., Suite 300 Boulder, CO 80301 USA
Blog	GET SOCIAL	Usage and Permissions	
	Twitter		
TRAINING	Linkedin		BUSINESS HOURS
	YouTube		Weekdays: 9am to 5pm Weekends: CLOSED
Course Calendar			
About Certification			
Become a Trainer			
			Privacy Policy
			Cookie Policy
			Your California

English

Scaled Agile Framework



It is hard to imagine a more stupid or more dangerous way of making decisions than by putting those decisions in the hands of people who pay no price for being wrong.

—Thomas Sowell

Business Owners

[Adjust Cookie Settings](#)

Business Owners (BOs) are key ART stakeholders who have the primary business and technical responsibility for return on investment (ROI), governance, and compliance.

Business Owners are critical stakeholders who evaluate fitness for use and actively participate in ART events and solution development.

Details

Self-managing, self-organizing [Agile Teams](#) and ARTs are essential to the success of SAFe. This Lean-Agile way of working represents a significant change in the traditional management mindset. Leaders and management no longer need to supervise work directly or assign tasks. Instead, they lead and provide intent by establishing a mission and [Vision](#).

Business Owners may help teams with coaching and skills development but essentially decentralize execution authority to the ART. However, transformation to a Lean-Agile way of working does not relieve leaders and management of their ultimate responsibilities. They remain accountable for the organization's growth and its people, operational excellence, and business outcomes. SAFe defines the role of BOs, the key leaders who guide ARTs to optimal business outcomes.

Questions to identify Business Owners include:

- Who is ultimately responsible for business outcomes?
- Who can steer this ART to develop the right solutions?
- Who can speak to the technical competence of the solution now and in the future?
- Who should participate in planning, help eliminate impediments, and speak on behalf of development, the business, and the customer?
- Who can approve and defend a set of PI plans, knowing they will never satisfy everyone?
- Who can help ARTs coordinate efforts with other departments and organizations, spanning organizational boundaries?

The answers to these questions will help identify the BOs who will play a vital role in the ART's ability to deliver value. Also, consider the availability of these leaders and their personal traits. Will they be an excellent [Lean-Agile leader](#)? Are they interested in fulfilling this role?

It's best to start with the smallest possible Business Owner team and then add members if it becomes clear that someone with the necessary accountability, skill, knowledge, or expertise is missing. Ensure a good mix of both business-oriented and technical people. It's a reasonable expectation that membership in the BO

team will change as needs dictate.

Responsibilities

An effective Business Owner is active and involved, fulfilling their SAFe responsibilities daily, as illustrated in Figure 1.

Figure 1. Business Owner Responsibilities

Figure 1. Business Owner Responsibilities

While there is no precise guideline about who should be part of the Business Owner team, they often have the following roles or titles:

- General or line of business manager
- Product or Solution Managers
- Enterprise Architects
- C-level executives
- Operations executives

Senior engineering leaders
Customers (for bespoke solutions)

The following sections describe the Business Owner's duties, enabling them to fulfill their obligations while empowering Agile Teams and trains to do their best work.

Leading by Example

Business Owners are [Lean-Agile Leaders](#) who share accountability for the business value delivered by a specific ART. The most important and effective technique for driving the cultural change needed for the adoption of SAFe is for leaders to internalize and model the behaviors and mindsets of [Business Agility](#). Such leaders inspire others to follow in their direction and to incorporate the leader's example into their development journey. To accomplish this, Business Owners:

Serve as an example of the new behaviors ▷ Live by the Lean-Agile principles and practices, modeling the new norms of expected behaviors for the ART and others to follow. They help address shortcomings in SAFe knowledge and experience.

Communicate the vision for SAFe adoption ▷ Frequently communicate the business need, urgency, and vision for change. BOs participate in developing the SAFe implementation plan, prioritizing the transformation backlog, and establishing the metrics for tracking the change progress for one or more ARTs.

Actively engage with the Lean-Agile Center of Excellence (LACE) ▷ Address problems that teams cannot resolve. Such issues are often beyond the span of control of the LACE. For example, they may require facility changes, funding, hiring, and purchasing authority.

Address the concerns of people who resist the change ▷ Exhibit empathy and compassion, address people's fears and worries, and resolve problems quickly and effectively to help overcome the resistance that may block the change.

Act as change agents ▷ Communicate passionately, sincerely believe, and illustrate their commitment to the future change vision. When people see leaders' behaviors modeling those required by change, they become change advocates, aligning with the new behaviors more quickly. BOs do not tolerate unacceptable behavior and inspire those who resist or fear the change with mission and vision. BOs help people understand the new way of working and how it will benefit them, other ART members, and the organization. These leaders assure people by committing to adapting roles, practices, and processes for the overall good of the organization and ART.

A lack of psychological safety at work can have significant business consequences. When people don't feel comfortable talking about things that aren't working, the organization is not equipped to prevent failure. After all, no one can fix a secret.

This fear often leads to disengaged employees and the *opportunity cost* to leverage the strengths of all its talent. People need to feel comfortable speaking up, asking naive questions, experimenting and failing with new ways of working, and disagreeing with changes to create and implement ideas that make a real difference.

In contrast, when employees are engaged, they adopt the organization's vision, values, and purpose. They become passionate contributors, innovating problem solvers, and dependable colleagues.

Engaging with LPM

While LPM is operated by executives responsible for business outcomes, Business Owners are often critically engaged in the process. Some Business Owners may serve as LPM executives, but most are involved to some extent in activities such as:

Strategy and Investment Funding □ Help ensure the portfolio and individual value streams are aligned and funded to create and maintain the solutions needed to meet business targets.

Agile Portfolio Operations □ Business owners are responsible for helping value streams and ARTs get the right thing out the door to their customers. They may also directly or indirectly support the LACE and foster **Communities of Practice (CoPs)** within their domain of concern.

Lean Governance □ Business Owners are directly engaged in ART backlog prioritization and value stream economics. They also help provide oversight and decision-making of spending, audit, compliance, forecasting expenses, and measurement for their value streams.

Occasionally serve as Epic Owners □ On occasion, Business Owners may serve as initial Epic owners for initiatives that benefit from their domain knowledge, experience, and authority.

Participatory Budgeting □ Business Owners actively assist LPM in allocating the total portfolio budget to its value streams.

Aligning Priorities and PI Planning

Business Owners are responsible for understanding and refining the **Strategic Themes** that influence ARTs. They have knowledge of the current **Enterprise, Portfolio, and Value Stream** context, and they're involved in driving or reviewing the solution vision and **Roadmap**. The continuous involvement of BOs during the PI serves as a critical **Guardrail** for the ART's budgetary spending. Aligning priorities and PI planning usually involve the following activities:

The time before PI planning is a busy period for Business Owners. Responsibilities include:

Provide input to backlog refinement ☰ Participate in activities to align the backlog with the portfolio's strategic themes

Ensure that business objectives are understood ☰ Ensure that the business objectives are agreed to by key stakeholders of the train, including the [Release Train Engineer \(RTE\)](#), [Product Management](#), [System Architects](#), and other BOs

Prepare to communicate the business context ☰ Prepare to describe the business's current state, the [Portfolio Vision](#), and their perspective on how effectively existing solutions address current customer needs.

The importance of the Business Owner's role during PI planning cannot be overstated. Activities include:

Present the business context and Vision ☰ Share the business context during the defined PI planning agenda timebox. This context may include the state of the business, market rhythms, milestones, and significant external dependencies, such as those of [Suppliers](#).

Actively engage during critical ART PI Planning activities ☰ Participate in draft plan reviews, assign business value to team PI objectives, and approve final plans.

Review draft and final plans ☰ Understand the bigger picture and determine if the team's objectives fulfill the current business objectives when taken together. They ask powerful questions and ensure alignment on solution intent.

Watch for significant external commitments and dependencies ☰ Foster the management of dependencies and support their reduction or elimination.

Actively circulate during planning ☰ Communicate business priorities to the teams and maintain agreement and alignment among the stakeholders regarding the key objectives of the train.

Participate in the management review and problem-solving ☰ Business Owners are critical stakeholders in this problem-solving meeting. They review and adjust the scope, resolve problems, and compromise as necessary.

Participate in Solution Train planning ☰ If applicable, BOs participate in [Pre-Planning](#), helping ARTs adjust their plans and providing support during the [Coordinate and Deliver](#) activities.

Moreover, when Business Owners assign planned business value during PI planning, it offers an essential face-to-face dialogue between teams and their most important stakeholders, the BOs. This activity is an opportunity to develop personal relationships between Agile Teams and BOs, identify common concerns that require mutual commitment, and better understand the business objectives and their value. Figure 2 provides an example of one team's PI objectives and the Business Value (BV) assigned by BOs.

Business Owners use a scale of one (lowest) to ten (highest) and will typically assign the highest values to the customer-facing objectives. However, they should also seek the advice of technical experts who know that architecture and other concerns will increase the team's velocity in producing future business value. So placing suitable business value on [Enablers](#) helps drive velocity and demonstrates their commitment to addressing the team's legitimate technical challenges.

Figure 2. An example of a team's PI objectives with assigned business value

SAFe customers often ask, why doesn't the BV use Fibonacci numbers? The answer is simple: the 10 to one scale is a range of numbers everyone understands, reducing friction and miscommunication between business-oriented BOs, technical members, and ART stakeholders. The simplest way to start is to assign a ten to the highest individual objectives, typically "fixed" commitments or must-have items, and then scale down from there. Giving many PI objectives a ten (for one team) indicates a lack of objective prioritization. It effectively abrogates prioritization to the team without the benefit of the BO's knowledge and experience.

Realizing Business Outcomes

The Business Owner's job is not complete when PI planning is done. They have an ongoing role in helping ensure the success of solution delivery. Business Owners typically:

- Maintain alignment** □ Actively maintain alignment between the business and development as priorities and scope inevitably change.

Help validate the definition of MVPs ☰ Guide pivot-or-persevere decisions for ART or Solution Epics based on the delivery of the MVP.

Attend the System and Solution Demos ☰ Actively engage in the system and solution demos to understand progress and provide feedback.

Attend Agile team events ☰ Attend team events such as [Iteration Planning](#), Review, and [Retrospectives](#) as needs dictate.

Actively address impediments ☰ Help resolve impediments that escalate beyond the authority of the trainOS leaders and stakeholders.

Participate in Release Management ☰ Serve as critical stakeholders in release governance (described in the [Release on Demand](#)) and determine when the solutions are released. Specifically, they focus on scope, quality, deployment options, release, and market considerations.

Sponsoring Relentless Improvement

The [Inspect and Adapt \(I&A\)](#) event is a cadence-based opportunity for the whole ART to reflect on progress and identify the systemic impediments they're facing—many of which require the BO's involvement. During the event, BOs assess the actual value achieved versus the plan and participate in the I&A's problem-solving workshop. Moreover, Business Owners are Lean-Agile Leaders who:

Continually focus on eliminating waste and delays ☰ Foster the adoption of Principle #6, [Make value flow without interruption](#), including [Value Stream Management](#), the eight flow accelerators, and the six flow measurements (*flow distribution, velocity, time, load, efficiency, and predictability*).

Eliminate demotivating policies and procedures ☰ Actively participate in the I&A's problem-solving workshop to identify and eliminate systemic issues, policies, and processes that are not aligned with the Lean-Agile Mindset and are not within the scope of the ART's control.

Inspire and motivate others ☰ Effectively communicate (frequently) *why* change is needed and do so in ways that inspire, motivate, and engage people to buy into the change with a sense of urgency.

Create a generative culture that highly values relentless improvement ☰ Model the right behaviors to help transform the culture from pathological (negative, power-oriented) and bureaucratic (negative, rule-oriented) to a positive, performance-oriented culture, which is required for the Lean-Agile mindset to flourish.

Provide the time and space for teams to innovate ☰ Foster the use of IP iterations to provide a regular, cadence-based opportunity for teams to work on innovation, improvement activities, and learning that are often difficult to fit into a continuous, incremental value delivery pattern.

Help drive investment in the continuous delivery pipeline ☰ Supports process and infrastructure enhancements to the [Continuous Delivery Pipeline](#) to improve the

responsiveness of the ART and the quality of its solutions.

It cannot be emphasized enough: Active participation of Business Owners is critical to the SAFe enterprise.

Last update: 6 February 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

- Weekdays: 9am to 5pm
- Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework

The Role of PI Objectives

by [Eric Willeke](#), SAFe Fellow, SPCT, [Elevate.to](#)

Note: This article is part of the [Community Contributions](#) series, which provides additional points of view and guidance based on the experiences and opinions of the extended SAFe [community of experts](#).

[Adjust Cookie Settings](#)

Introduction

The role of PI Objectives is often misunderstood by teams new to PI Planning. They struggle at first to understand the difference between Team PI objectives and Features. SAFe does not provide a lot of guidance on the intent behind the usage of PI Objectives, and they are often misunderstood or misinterpreted. However, in my field practice, I've come to really value them, so I wanted to take the time to document my perspective in this article.

The main qualities of PI Objectives that I have come to value are their ability to:

- Validate understanding of the intent
- Focus alignment on outcomes rather than process
- Summarize data into meaningful and steerable information

Without understanding the above qualities, it's quite easy to view PI objectives as nothing more than a shorthand list of Features to be delivered.

Validate Understanding of Intent

Two of SAFe's four core values are Relentless Improvement and Alignment, and one of the key Lean principles underlying SAFe is decentralized decision-making. Most of the PI Planning event agenda is focused on supporting these values by clearly conveying the desired goals and outcomes for the Agile Release Train, then getting out of the way and enabling practitioners to achieve these goals effectively. We place governing structures and other feedback loops into play through the product manager and Product Owner roles, but we generally support the teams' ownership of the details in pursuit of the larger business outcomes.

However, one of the key risks that have plagued every software development approach is ensuring that the initial intent (scope) was clearly understood and articulated by the stakeholder, transmitted effectively to the development team, and interpreted in the same way. In SAFe, this path also includes the extra steps of translating from the end users to the Business Owner (who requires this capability), then onward to the product manager (for the release train implementing the capability.)

SAFe's use of PI Objectives provides a unique tool to create an immediate feedback loop from the teams back to the Business Owners, allowing a quick validation of the teams' grasp of the desired outcomes. In short, we give the teams the following challenge:

“Can you concisely convey, in words the business owner understands, the essence of the value sought by implementing this set of features?”

If a team cannot do this in a clear way by the end of planning, are we comfortable investing over \$100,000 to pursue these goals over the next 10 weeks? By forcing the teams to summarize the intent and the outcomes they believe the Business Owner wants to achieve, we close the loop of understanding and drive crucial conversations that expose these misunderstandings. This, in turn, enables a much tighter form of alignment that transcends the written language of the feature to be amplified by the tacit understanding gained between the team and the Business Owner.

Focus on Outcomes Rather than Process

The second hidden value of PI Objectives is that they help the team shift focus off the feature language and onto the desired business outcomes. Features and acceptance criteria are amazing tools to help understand, capture, and collaborate around the work that needs to be done to iterate our solution to the next level, but it's all too easy to get caught up in "finishing the features" and missing the overall goals hiding inside of them. The core question becomes:

Is our goal to complete the listed features, or is our goal to provide the outcomes desired by those features? In other words, if we could provide the same value with half the amount of work, and without building all of the features, would this be acceptable?

I've found that the language of features can frequently steer the team into overlooking creative, valid, and architecturally sound solutions because someone outside the team has already provided a preconceived notion of how that value should be provided. The closer understanding of the intent offered by direct conversations with the Business Owner occasionally results in the teams offering new perspectives to the architects and product managers and quickly finding ways to apply their expertise more effectively.

Summarize Data into Steerable Information

Finally, there's a simple "comprehension" aspect to PI Objectives that I find particularly valuable. I've come to accept that no large group will reliably read every item in a list if that list exceeds 5-7 items. Given that I see small trains with only four teams consistently take on 10 features per PI and large trains that have taken on as many as 40, I assume nobody except the product manager reads every single feature carefully—and certainly, nobody outside the train has read every one. As a result, I deeply value the summary of intent that the ART PI Objectives provide and the subsequent use of those objectives as a key for providing clear evidence of progress both within and beyond the train.

I strongly encourage teams to fully and transparently share the features they intend to complete and their progress against them in a percent of a complete mindset. But I've also found it valuable to summarize the 5-7 key objectives per train and report on progress against those. This is especially true when you have four or more trains working against the same value stream, aggregating a shared system demo to a senior executive audience every two weeks. Quite simply, you need a more compact way to convey the same information to augment quantitative reporting.

A Bit of Philosophy

As I captured these thoughts, I was reminded of a key difference between release trains, and it strongly impacts the degree of value placed on the "focus on outcomes" above. I tend to see release trains fall to one of two extremes: Either they drive the vast majority of their work (85-95%) through portfolio epics and reduce the autonomy of the trains, or they drive the vast majority of their work as features, and reserve epics for the 5-10% of work that truly cuts across trains. While I don't view either of these approaches as fundamentally wrong, I do believe that we will see that a majority of companies solving the big "system of systems" engineering problems, encouraging a strongly epic-driven approach, will find that SAFe for Lean Software and Systems Engineering will be a more effective approach.

At the same time, I fundamentally believe that the mindset of epic-driven development is serving the same role for steering at the scale that "waterfalling iterations" served for teams: it provides a way for organizations to avoid learning those crucial lessons about how to work together in ways we never had to in the past.

Last update: 5 May 2020

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Strive to be more Agile, rather than simply following Agile methods and steps. This approach encourages adoption of the philosophy, or mindset, rather than specific steps. This is also referred to as being Agile, or having agility versus using it.

Adjust Cookie Settings

Ñ Effective Practices
and Federal
Challenges in
Applying Agile
Methods, GAO, 2012
[1]

Building a Solid Lean-Agile Foundation

This is article one in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

One of the more common anti-patterns of Agile adoption is the misconception that simply following methods such as Scrum will lead to development becoming ÔAgile.Õ As a result, many previous attempts to incorporate Agile methods in government agencies have struggled and havenÕt resulted in increased frequency and quality value delivery. Root cause analysis often reveals that leaders assumed they would generate Agile benefits by merely adopting practices such as working in two-week sprints, organizing work as stories in a backlog, and conducting daily stand-ups.

Achieving mission agility through technology development, however, requires more than checking off a list of practices from any of the common team-level Agile methods. Instead, it begins with understanding *why* Lean-Agile produces better software and systems development results versus waterfall. (For example, Lean-Agile is 350% more likely to be successful compared to waterfall projects, and 600% more likely for very large projects according to the Standish Group [2]). It requires addressing issues of mindset and culture, which are admittedly more difficult to change than the mechanics of organizing and executing work. In fact, for more than a decade, the annual State of Agile Report has listed Ôorganizational culture at odds with Agile valuesÕ among the top challenges in adopting and scaling Agile. [3]

Fortunately, the bodies of knowledge of Lean, Agile, and SAFe all have foundational values and principles that guide us to the right perspectives needed to execute a successful transition from traditional lifecycle models. The sections that follow provide brief summaries of the values and principles of Lean-Agile and SAFe, with links to the detailed articles in the Framework that provide further elaboration.

Details

Lean Thinking

The core benefit of Lean is to maximize customer value while minimizing waste. The original principles of Lean include:

- Precisely specify value by product
- Identify the value stream for each product
- Make value flow without interruptions
- Let the customer pull value from the producer
- Pursuing perfection [4]

It is easy to see how Lean thinking applies in the government context. Maximizing value delivered for taxpayer money invested in technology while minimizing wasteful spending should be a universal goal of every public agency. Unfortunately, traditional development practices do not sufficiently emphasize or enable these objectives. Further, the ideas of respect for people and culture, flow-based development, investing in innovation, and relentless improvement have only recently gained acceptance in federal software and systems programs thanks to resources such as the Digital Services Playbook. [7] Understanding the principles and practices of Lean is a critical prerequisite to applying these concepts for better technical performance and mission results.

Agile Values and Principles

Many in government don't associate the term "Agile" with a set of values and principles. Instead, they picture small teams conducting daily stand-ups in front of marker boards full of Post-It™ notes. Few envision a fundamental and philosophical shift emphasizing:

- Face-to-face conversations
- Evaluating progress by observing working components versus reports
- Collaborating with customers to define and validate system functionality
- Having a development plan that can deal with a dynamic environment

Yet, that is precisely what is needed in public sector technology programs. No one has articulated this more eloquently than Gen. Ellen Pawlowski (Ret.), former Commander of US Air Force Materiel Command. (See Gen. Pawlowski's explanation of how the 12 principles of Agile directly and immediately apply to mission accomplishment in the US Department of Defense here:

<https://youtu.be/nQUplJVjql>).

Following a wave of infamous software project failures in the 1990s, a group of technology thought leaders articulated Agile values and principles in 2001. [8] As compared with the traditional plan-driven model that had been the default standard since the early 1970s, these experts envisioned a more iterative and incremental customer-focused approach. The values of the Agile Manifesto describe a transformation in the approach to developing software. They include:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

The four values are supported by 12 principles that create a tangible picture of how an Agile approach changes the way development teams work. These principles can be found on the Agile Manifesto website [9] as well as in the second half of the [Lean-Agile Mindset](#) article.

SAFe Values and Principles

The Scaled Agile Framework (SAFe) is firmly grounded in, and consistent with, the values, principles, and practices of Lean and Agile. SAFe emerged a few years after the Agile Manifesto as the world's most prominent organizations attempted to apply Lean and Agile to the development of large and complex products. These included core banking software, electronic medical equipment, logistics and supply chain systems, and fighter jets. The experiences of these enterprises revealed that at scale, an additional set of values and principles for technology development are essential to produce high-value quality products in the shortest sustainable lead time. The four SAFe core values and nine principles are illustrated in Figure 1.

Figure 2. The SAFe Core Values and SAFe Principles

Figure 1. The SAFe Core Values and SAFe Principles

Over the last 10+ years of SAFe implementations in Global 1000 companies and governments around the world, the most successful organizations embrace and embody these values and principles. Conversely, in instances where there have been problems in a SAFe transformation, the most common root causes can be traced to the failure to observe and enforce one or more of these key tenants. Dozens of successful SAFe implementations in government agencies in the U.S. and abroad have also proven that these values and principles are equally applicable and achievable in the public sector.

For a more detailed discussion, refer to the articles on [Core Values](#) and [SAFe Principles](#).

Critical Success Factors

Finally, in addition to a solid foundation of values and principles, we've found that

a core set of 10 critical success factors are vital to any SAFe implementation. Although the majority of the framework is configurable and designed to be tailored to each specific context, the most successful implementations have consistently embraced the SAFe's [ten critical success factors](#) shown in Figure 2.

Figure 3. The 10 critical success factors of SAFe

Figure 2. The 10 critical success factors of SAFe

That article provides a detailed explanation of each of these fundamental SAFe practices, along with links to toolkits that can be used to present them and assess how well an organization is doing in following these core practices.

Government leaders who commit to the values and principles of Lean, Agile, and SAFe—along with the 10 critical success factors—when managing technology initiatives, can expect immediate and measurable improvements. These include increases in program predictability, quality, the frequency of value delivery, customer satisfaction, and positive employee and supplier engagement.

Moving Forward

The next crucial concept for adopting SAFe in Government is **Creating high-performing teams of teams.**

NEXT

Learn More

- [1] *Effective Practices and Federal Challenges in Applying Agile Methods.* GAO, 2012. <https://www.gao.gov/assets/600/593091.pdf>
- [2] *Chaos Report.* Standish Group, 2015. <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
- [3] *16th Annual State of Agile Report.* <http://www.stateofagile.com>
- [4] Womack, James P., Daniel T. Jones, and Daniel Roos. *The Machine That Changed the World.* Free Press, 1990.
- [5] Reinertsen, Donald G. *The Principles of Product Development Flow: Second Generation Lean Product Development.* Celeritas, 2009
- [6] Poppendieck, Mary, and Tom Poppendieck. *Implementing Lean Software Development From Concept to Cash.* Addison-Wesley, 2006.
- [7] The Digital Services Playbook. <https://playbook.cio.gov/>
- [8] To Agility and Beyond: The History & Legacy of Agile Development. <https://techbeacon.com/agility-beyond-history%E2%80%94-legacy%E2%80%94-agile-development>
- [9] The Agile Manifesto. <http://AgileManifesto.org/>
- [10] Gen. Ellen Pawlowski, US Air Force Materiel Command. *DoD Use of Agile Software Development* *Critical.* <https://youtu.be/nQUppIJVjql>

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



We get together on a frequent basis, compare notes about what the problems were and what anybody could do to help the situation. And there was a spirit of cooperation pretty much throughout the program. At no point was any team in the dark about what another group was doing, or what

Adjust Cookie Settings

*support they
needed.*

ÑOwen Morris, Chief
Engineer of the
Lunar Module,
Apollo Space
Program [1]

Creating High Performing Teams of Teams

This is article two in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

The concept of cross-disciplinary teams of government and contractor personnel working cooperatively to create complex technological systems is not new. Perhaps the most famous illustration of the history of government-industry collaboration is the Apollo space program. That decade-long effort involved over 400,000 engineers, scientists, and technicians from government and from more than 20,000 commercial companies and universities. [1] These teams

Shared a vision and focus

Were willing to cooperate with all other teams regardless of their organization

Solved problems in an open, collaborative fashion

That team-centric environment was put to the ultimate test during the near-tragic Apollo 13 mission.



Figure 1. Engineers from multiple organizations responding to the Apollo 13 mission crisis

In more recent years, the use of technology has exploded to become a mission-critical capability for every function in every government agency. Unfortunately, that Apollo spirit has devolved in some instances into an environment of low trust Øus vs. themØ behaviors between government and contractors. Additionally, in an era of hyper-competition for limited contract awards, multiple suppliers on the same program sometimes work to undermine each other in hopes of unseating the competition in the next re-compete. Finger-pointing and blame game responses to defects and shortcomings frequently overshadow opportunities to collaboratively problem solve for the benefit of the ultimate stakeholder of government systemsÑ the taxpaying citizen.

The first article in this series made the case for establishing a firm foundation in Lean-Agile and SAFe values and principles as the key to faster value delivery in the sustainably shortest lead time. However, these ideals do not function well in low trust environments. If government agencies are going to realize the promise of better mission results through iterative and incremental development, a focus must be applied to building the same type of healthy communication and collaboration among the people creating complex government systems that once achieved the national vision of putting a man on the moon.

Details

Defining High Performing Teams

ØThe performance challenges that face companies in every industry Édemand the kind of responsiveness, speed, online customization, and quality that is beyond the reach of individual performance. Teams bridge this gap.Ó

Ñ Jon Katzenbach & Douglas K. Smith, *The Wisdom of Teams* [2]

The Center for Organizational Design describes high performing teams as Óa group of people who share common vision, goals, metrics, and who collaborate, challenge and hold each other accountable to achieve outstanding results.Ó [3] They further elaborate that high performing teams:

- Have a clear and vision of where they are headed and what they want to accomplish
- Are excited about that vision because they took part in creating it
- Act from clearly defined priorities
- Have clear measures of success and receive feedback about how theyÕre doing
- Maintain open communication and positive relationships with each other
- Identify and solve problems
- Make decisions when and where they occur
- Successfully manage conflict
- Share leadership responsibilities
- Participate in productive meetings
- Have clearly defined roles and work procedures
- Cooperate cross-functionally

Even when teams are a blend of government employees and practitioners from multiple contractors, government technology programs should consist of high-performing teams that display these attributes.

Combined with technical best practices for modern systems development, high-performing teams produce mission results that are significantly better than their low-performing counterparts. According to the 2017 State of DevOps Survey [4], high-performing teams:

- Experience code deployments occur 46 times more often
- Lead time from commit to deploy is 440 times faster
- Meantime to recover from downtime is 96 times faster
- Change failure rate is 5 times lower with those changes being 80% less likely to fail

Achieving these results requires creating an environment where high-performing teams can emerge and thrive. Agencies should include language expectations in each industry partnerÕs contract for how contractors and government personnel will work collaboratively in the best interests of the mission. Contracts should describe how the government expects various contractors working on the same program to cooperate with each other. Also, government leaders create the environment for high-performing teams by how they and their civil service staff

interact with contractor team members to create a generative trust-based workforce culture.

Building Healthy Agile Teams

Healthy Agile teams are the primary organizational unit in SAFe. They have many of the attributes of high-performing teams built-in by design. For example, Agile teams in SAFe are small, cross-functional, and self-organizing. They have the trust of the organization to define and execute the work needed to accomplish the objectives of the program. Their [Product Owner](#) gives them clear priorities, and they have clear acceptance criteria for each element of functionality they build. They work in single piece flow from a backlog of user and enabler stories that align to a clear and common vision. Each [PI](#) Agile teams dedicate time to engage in problem-solving and innovation toward the goal of relentless improvement. They commit to the work they plan every PI and are accountable for predictability over time, measured by consistent delivery of those commitments.

That's what a healthy Agile team looks like. Anything less falls short of the full potential of Lean-Agile adoption. Successful Agile programs in government have proven that the goal of developing high-performing government and contractor Agile teams is very achievable ([Case Studies](#)).

Click here for additional details regarding the composition and responsibilities of [Agile teams](#).

Forming Multi-Organizational Teams-of-Teams

“Today’s rapidly changing world, marked by increased speed and dense interdependencies, means that organizations everywhere are now facing dizzying challenges, from global terrorism to health epidemics to supply chain disruption to game-changing technologies. These issues can be solved only by creating sustained organizational adaptability through the establishment of a team of teams.”

— Gen. Stanley McChrystal (Ret.), *Team of Teams: New Rules of Engagement for a Complex World* [5]

Figure 2. Navy SEALS and other special operations organizations are examples of teams of teams

Few government technology programs can be delivered by a single Agile team. To build complex systems like global GPS networks and veterans eligibility systems, many teams are required. They must maintain all the elements that distinguish Agile teams while working cooperatively for prioritization, integration, and dependency management. They must plan together, work together, integrate together, demo together, innovate together, and improve together. Without a clear strategy to reinforce these collaborative activities, it becomes too easy for vital communications to become siloed and to fall victim to the tyranny of the urgent.

In the context of high speed, variability, and interdependency, the best organizational structure is a “team of teams.” [5] In SAFe, the team of teams is an Agile Release Train or ART. ARTs have many of the same attributes as Agile teams at a fractal-level larger scale. Most government programs are large enough to form one or more ARTs.

Click here for additional details on [Agile Release Trains](#).

Integrating Between Multiple ARTs and Suppliers

Some government technology programs are so large that they involve hundreds or even thousands of practitioners. Examples include Vets.gov, the F-22 fighter jet, and France’s centralized unemployment system. These solutions require far more than a single team-of-teams; they can require many ARTs to build these complex systems-of-systems. It is also common for these large programs to include large numbers of suppliers. Yet all of these practitioners and organizations must work together to build, integrate, test, and deploy a single unified solution.

Figure 3. Large Solutions require many ARTs and Suppliers working together as a Solution Train

SAFe provides extensive guidance on how to plan, coordinate and manage programs of this size and complexity. To learn more, read [Large Solution SAFe](#) and the [Enterprise Solution Delivery](#) articles.

Defining Clear Roles and Responsibilities

Organizing the product development workforce into Agile teams and trains requires roles and responsibilities that are different than traditional top-down organizational chart-based models. For example, healthy Agile teams have moved away from being directed by a team leader responsible for hiring, planning, tasking, prioritization, and performance evaluation. In Agile, the traditional leader model has been turned upside down; the responsibilities are spread among the team and to new roles such as Product Owner and Scrum Master/Team Coach. Team members still report to people managers, who now have more focused

responsibilities for hiring, promoting, and facilitating the growth of the practitioners within the teams. Other leaders outside the ART still support the product development process by using their authority to provide the vision, resources, decisions, etc. that allow teams to be successful. A pattern of cooperative collaborations replaces the conventional top-down command-and-control model, with servant leaders using their position and influence to ensure team and ART success.

Figure 4. ARTs are virtual organizations characterized by cooperative collaborations

SAFe provides detailed descriptions of the roles, responsibilities, key activities, and collaborations at every level within the Framework (Essential, Large Solution, and Portfolio). These descriptions can be found in the following articles:

Essential → [Agile Teams](#), [Product Owner](#), [Scrum Master/Team Coach](#), [Product Management](#), [System Architect](#), [Release Train Engineer](#), [Business Owners](#)

Large Solution → [Solution Management](#), [Solution Architect](#), [Solution Train Engineer](#)

Portfolio ☰ [Lean Portfolio Management](#), Enterprise Architect, Epic Owners

It is important to note that all of these roles describe specific responsibilities to be performed in a *virtual* SAFe organization, performed by personnel with a wide variety of job titles. Likewise, government agencies should use broad discretion in the labor categories used to fulfill these responsibilities if the roles are performed by contractor staff. This provides maximum flexibility and cost management for both parties while ensuring that practitioners with the right skill sets are performing each of these roles. Significant pre-planning is also recommended regarding which roles are best performed by government personnel versus those that can be filled with government or contract staff.

Finally, many agencies have unionized personnel who will be asked to perform new roles at various levels in a SAFe implementation. The best advice from agency executives who've dealt with this issue is to be as proactive as possible, communicating with union representatives about the changes in roles that come with shifting from legacy waterfall practices to SAFe. There are tangible benefits in the form of additional training, certifications, and career growth that make a SAFe transition a win-win for agencies and their union workforce. The key is to create an open dialogue for shared understanding, addressing concerns early and avoiding serious impediments to the implementation roadmap.

Moving Forward

The next key concept for adopting SAFe in Government is **Aligning technology investment with agency strategy**.

NEXT

Learn More

[1] How collaboration put a man on the moon.

<https://medium.com/@zeroagency/how-collaboration-put-man-on-the-moon-d21f9fb3ce8c>

[2] Katzenbach, Jon R., and Douglas K. Smith. *The Wisdom of Teams*. Harvard Business Review Press, 1992.

[3] *Developing high performing teams.* The Center for Organizational Design.
<http://www.centerod.com/developing-high-performance-teams/>

[4] *2017 State of DevOps Report.* Puppet + DORA.
<https://puppet.com/resources/whitepaper/state-of-devops-report>

[5] McChrystal, Stanley (Retired General), et al. *Team of Teams: New Rules of Engagement for a Complex World.* Penguin Publishing Group, London, England, 2015.

Last update: 5 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK	PARTNER	CONTENT & TRADEMARKS	SCALED AGILE, INC
Download SAFe Posters & Graphics	Becoming a Partner	FAQs on how to use SAFe content and trademarks	CONTACT US
Watch and download SAFe videos and presentations	Partner Directory	Permissions Form	5400 Airport Blvd., Suite 300 Boulder, CO 80301 USA
Blog	GET SOCIAL	Usage and Permissions	BUSINESS HOURS
TRAINING	Twitter		Weekdays: 9am to 5pm Weekends: CLOSED
	Linkedin		Privacy Policy
	YouTube		Cookie Policy
			Your California

English

Scaled Agile Framework



*However beautiful
the strategy, you
should occasionally
look at the results.*

Ñ Sir Winston
Churchill

Aligning Technology Investment

[Adjust Cookie Settings](#)

with Agency Strategy

This is article three in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

Technology investments are a vital part of each government's annual budget. Increasingly, modern information technology has been identified as the backbone for how government serves and meets the expectations of citizens. As a result, each agency has to articulate the initiatives and the funding required to implement their piece of the overall management agenda. The question is—how much of the technology budget (including cyber-physical systems) in each agency is aligned to the government's larger aims?

This article explores how the leaders responsible for managing an agency's technology investment portfolio can use the tools within SAFe to create alignment between the Agile Release Trains (ARTs) in their development organization and the strategies of the agency and the federal government.

Details

Challenging Status Quo



If you always do what you've always done, you'll always get what you always got.

— Henry Ford

The U.S. federal government budget for 2022 included \$109.4 billion for information technology (IT) [1] and over \$245 billion more on weapons systems, satellites, and other physical systems. [2] Other governments also invest heavily in technology as a percentage of their total budget. In an era of unprecedented transparency and scrutiny over how governments spend taxpayer dollars, it is essential these investments deliver the highest value. Additionally, many of these systems literally protect the lives of the general public. However, too often, government agencies worldwide adopt a mindset of what Eric Ries (author of *The Lean Startup* [3] and *The Startup Way* [4]) refers to as “entitlement funding.” This assumes a never-ending spigot of money to support systems that may be antiquated, redundant, or no longer aligned to the agency's strategy. Entitlement

funding also encourages what Ries calls “success theater”—carefully selected metrics chosen to prove these projects and programs are “on target,” and thus merit additional investments made year-over-year, *regardless of their contribution or efficacy.*

The alternative to entitlement funding is to challenge status-quo thinking, to continuously evaluate whether technology investments are aligned with agency priorities. In the spirit of adopting a Lean mindset, one important exercise is to identify, reduce or eliminate waste as part of an agency’s technology budget. For example, it’s common for an agency to have multiple disparate systems performing duplicated functions. Eliminating wasteful spending was listed as “priority number one” in the U.S. 2019 budget. First on the list in the administration’s management agenda in support of that priority is modernizing outdated and redundant technology. Agencies, in turn, reflect these priorities in their own plans. For example, goal #4 in the 2018-2024 strategy document for the Department of Veterans Affairs is to modernize its technology systems. [5]

For an illustration of how IT modernization and application rationalization led to \$9 billion in savings in fiscal 2018, read the FedTech article [here](#). [6]

Objectives such as modernization or improved customer experience through digital services will exert considerable influence on the people, programs, and architectures needed to realize them. Approaches such as building autonomous platforms to replace manned-vehicles, moving applications to cloud platforms, or replacing custom applications with commercial off-the-shelf (COTS) systems can dramatically affect how an agency’s technology dollars are spent. They also determine what skills will be needed and how people and resources will be organized.

Another approach that is being adopted by the U.S. federal government to understand and control technology spending is Technology Business Management (TBM). TBM is a framework of practices that seek to create a true “bill of IT” so that government leaders can make informed, data-driven decisions on aligning technology expenditures to agency strategy. Agencies that have adopted both SAFe and TBM report strong synergies between the two bodies of knowledge. (For more details on TBM and SAFe, read the white paper [Accelerating Business Value with SAFe and Technology Business Management](#)).

If the strategy paints one picture of the future, but the department’s development budget conveys something else, the lack of alignment will undermine the organization’s ability to accomplish its mission. A different approach is required.

Creating Alignment with Strategic Themes

The primary mechanism in SAFe designed to address this challenge is [Strategic Themes](#). Agency leaders use strategic themes as the bridge between the strategy of the organization and its portfolio of technology initiatives, as shown in Figure 1 below.

Figure 1. Strategic themes connect agency strategy to technology funding and priorities.

Strategic themes *translate* agency imperatives into language that articulates the implications of each strategy on the organization's technology development efforts. Understandable and actionable strategic themes guide the people in the agency's development [Value Streams](#). Continuing the technology modernization example, here is an illustration of how federal and agency strategy can be expressed to the development organization as one or more strategic themes.

Figure 2. Example of Strategic Themes creating alignment with agency and federal strategies

Strategic themes are then used by the executives responsible for development to create a vision of the systems needed in the agency's technology portfolio. That vision influences budgeting decisions and informs a prioritized list of large initiatives ([Epics](#)) for one or more [Agile Release Trains](#) to implement. This, then, aligns the work of technology practitioners with the top priorities of the agency. SAFe recommends using Objectives and Key Results (OKRs) to define strategic themes, as it provides an effective means to define, organize and communicate these critical differentiating mission objectives.

For a more detailed discussion of this topic, please read the [Strategic Themes](#) and [OKRs](#) articles.

Building a Lean Portfolio Management

Competency

Aligning the efforts of the development organization with the overarching strategies of the agency is just one element of developing Lean Portfolio Management (LPM) as a core competency. SAFe provides guidance and tools, such as the portfolio canvas and participatory budgeting to support LPM activities. For a more in-depth discussion of this competency, read the [Lean Portfolio Management](#) competency article. You can also explore the associated roles, processes, and artifacts by clicking on the various icons in the [Portfolio](#) layer of the SAFe Big Picture. (Be sure to select the Full SAFe or Portfolio SAFe configuration from the website.)

Moving Forward

The next key concept for adopting SAFe in Government is **Transitioning from projects to a Lean flow of Epics.**

NEXT

Learn More

[1] Budget of the U.S. Government. Fiscal Year 2022.

<https://www.govinfo.gov/app/collection/budget/2022>

[2] Program Acquisition Cost by Weapons System. 2022 Fiscal Budget Request, U.S. Department of Defense.

https://comptroller.defense.gov/Portals/45/Documents/defbudget/FY2022/FY2022_Weapons.pdf

[3] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, 2011.

[4] Ries, Eric. *The Startup Way: How Modern Companies Use Entrepreneurial Management to Transform Culture and Drive Long-Term Growth*. Crown Business, 2017.

[5] Department of Veterans Affairs FY 2018 – 2024 Strategic Plan.

https://www.jcs.mil/Portals/36/Documents/Doctrine/Interorganizational_Documents/dva_

[strategicplan2018_2024.pdf](#)

[6] *How DHA and USDA Have Cleaned Up Their Cluttered App Portfolios.*

FedTech, July 23, 2018. <https://fedtechmagazine.com/article/2018/07/how-dha-and-usda-have-cleaned-their-cluttered-app-portfolios>

Last update: 5 October 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



The value is in what gets used, not in what gets built.

— Kris Gale, Clover Health (formerly with Yammer)

Transitioning from Projects to a

[Adjust Cookie Settings](#)

Lean Flow of Epics

This is article four in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

Traditional “project management” processes for technology development are deeply ingrained in government policy, procedures, and tools. Guidelines for project governance often suggest a waterfall, phase-gate lifecycle model. However, what we’ve learned from decades of challenged projects is that the nature of technology development is inherently empirical, requiring an iterative and incremental process similar to the scientific method commonly found in R&D labs.

This differs from the defined process model of planning all the work in detail at the beginning of the project. To get better results, we need to place greater focus on the value delivered versus the meticulous planning of tasks, especially early on when the cone of uncertainty is the greatest. This represents a significant shift in how government leaders might think about their technology initiatives. The good news is that once the mental hurdles are cleared, the practices for iterative product development are well defined, and are even recognized as a best practice in the GAO Agile Assessment Guide [1].

This article describes the shift from *organizing technology efforts as projects* to a *continuous flow of product-centric features in a value stream*. It provides best practices for managing that flow using a SAFe construct called “Epics”.

Details

Move from Projects to Products using Epics

Historically, government agencies have developed and maintained mission-enabling technologies through a sequence of start-stop projects. For new systems, the first “greenfield” project provides the initial operating capability (IOC), while subsequent projects deliver ongoing operations, maintenance, and enhancements of the system. Miscellaneous ancillary projects are also needed to field the new capability. These projects provide:

- Technical infrastructure
- Security (DevSecOps)
- Independent validation and verification (IV&V)
- Training and fielding

Systems engineering and technical assistance (SETA)

Each project typically starts with a ramp-up effort, followed by a series of projects often referred to as the “sustainment tail.” These disconnected initiatives and their associated contracts (and re-competes) make it difficult to holistically manage the costs and ongoing effort associated with each discreet capability.

Additionally, the traditional project management process assumes a waterfall approach where all of the requirements are defined upfront, followed by all the design, all the development, all of the testing, and finally the deployment of the finished product.

The flaw in this model is that most modern technology initiatives have too many unknowns in the beginning to develop accurate, detailed plans and designs upfront. Decades of failed projects prove that a different approach is required.

These challenges can be solved, in part, by shifting from building capabilities through disparate start-stop projects that manage tasks, to a holistic *product* management paradigm that encompasses all the efforts and resources needed to field and maintain a long-lived capability. The collection of people (government and contractors), systems, information, and material needed to provide a mission-enabling capability is referred to as a *Value Stream*. A value stream can contain one or more products (or systems) that work together to provide a capability.

Figure 1. A Fighter Jet capability requires a value stream of multiple complex systems.

The example in Figure 1 shows a fighter jet as the Solution produced by a value stream—a collection of products (in this case, systems) that include airframe, propulsion, avionics, communications, navigation, targeting, safety, and many more. Collectively, they provide a technological ecosystem that delivers a critical capability to support the missions required for air superiority.

Instead of organizing the work in a value stream into dozens of disconnected projects, SAFe’s Lean-Agile product approach uses epics—large initiatives, whose cost and scope warrant analysis, ROI assessment, a lean business case, and approval. Epics are visualized and managed in flow using Kanban and the principles of Lean thinking, as described in the [Lean-Agile Mindset](#), [SAFe Principle #6 Make Value Flow without Interruptions](#), and [Portfolio Flow](#) articles.

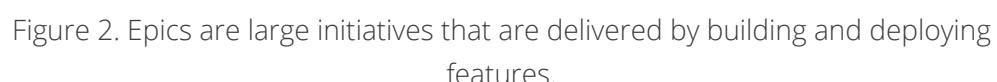


Figure 2. Epics are large initiatives that are delivered by building and deploying features.

As Figure 2 illustrates, Epics are in turn decomposed into [Features](#); these represent the new functionality built, deployed, and maintained by one or more [Agile Release Trains \(ARTs\)](#). Epics typically take more than one [PI](#) to deliver, but ideally no more than two to four PIs to keep batch sizes small (even at this larger scale).

They provide more frequent opportunities to deliver value, to learn, and to change the approach if necessary. Epics also contain all of the work (development, testing, IV&V, infrastructure, etc.) needed to provide a capability that was previously contained in numerous unrelated projects.

In the fighter jet example, an epic might be required to add functionality to use a newly developed missile. That enhancement could:

- Necessitate changes to multiple software, firmware, and hardware components
- Involve many teams over several PIs

Call for a significant investment

Epics in SAFe provide the appropriate planning, budgeting, and governance elements to deliver this modification in flow using Lean practices.

Commission Cross-Domain Teams for Rapid Prototyping

Once an agency adopts epics to provide a Lean alternative to projects, the next question is which epics to invest in, given time and budget constraints for new development. A Lean best practice is to perform rapid prototyping using cross-domain teams to explore alternatives for new or enhanced capabilities *before* large investments are made. In SAFe, the pattern for these exploratory prototypes is modeled after the work of Eric Ries in *The Lean Startup* [2] and *The Startup Way* [3]. Small cross-domain “two-pizza” teams are given specific time and cost parameters to evaluate alternatives and establish feasibility through rapid experiments. For accountability, multiple oversight committees converge into a single review board (described by Ries as a “growth board”). This combined review board also makes the final decision on whether to pursue the best alternatives that emerge or pivot to different strategies based on validated data from the experiments. For more details on how to continuously explore future capabilities and align investment decisions to Lean budgets, read the [Agile Product Delivery](#) and [Lean Portfolio Management](#) competency articles.

Apply the Lean Startup Cycle to Epic Development

One of the most valuable shifts in thinking provided by moving from projects to products using epics is the ability to deliver only those features that provide the required value. This reduces wasteful spending and avoids investing in unnecessary functionality.

Research by The Standish Group in 2002, updated in 2014, shows that 80% of the features in custom application development are used infrequently if ever. And only 40% of successful software projects (those that met the triple constraint of time, cost, and scope) delivered high or very high value. [4]

Citation:

https://www.standishgroup.com/sample_research_files/Exceeding%20Value_Layout.

[pdf](#)

In SAFe, this is accomplished by decomposing epics into features, and then prioritizing them in a backlog so that the next most valuable feature (greatest mission value for the smallest job size) is always completed next by teams on one or more ARTs. Once the Minimum Viable Product (MVP – the smallest set of features that collectively meet the mission requirements) is delivered, agency leaders can determine if the remaining features are worth the additional investment. Or, they can decide that those development funds would be better spent elsewhere.

Figure 3. Use the Lean Startup cycle to deliver only the features that meet mission requirements, reducing waste.

As an example, the epic to support a new missile on a fighter jet might be decomposed into 100 features, of which the first 70 in priority order would represent the MVP (i.e., the minimum required to achieve the mission). Once the first 70 features are built, tested, and deployed, agency leaders have full control

over how much additional investmentÑvia the remaining 30 featuresÑis required. If the decision is made to deliver an additional 10 and then stop, the 20 lowest priority features simply would not be built. The agency, and taxpayers, would then save the cost of developing and maintaining those 20 potentially wasteful features over the entire useful life of the system!

Traditional projects do not provide an equivalent mechanism. Once the time, cost, and scope are established, programs and their contracted industry partners are contractually and financially incentivized to deliver everything in the scope, regardless of the real value of the lowest priority features.

Visualize the Flow of Value

At any given time, an agency will have multiple epics proceeding through various stages of the lifecycle process. To achieve Lean flow, itÕs important to be able to visualize all of the work-in-process (WIP). For that reason, SAFe embraces the best practices of Kanban, supporting Lean disciplines such as enforcing WIP limits to enable sound decision-making and avoid overloading development capacity. This helps manage expectations while making development commitments visible and predictable. It also facilitates collaboration among key stakeholders as facts change, allowing trade-off decisionsÑsuch as prioritization and resource allocationsÑbe made transparently and quantitatively.

Figure 4 below provides an example of how an agency might visualize the flow of epics using Kanban.

Figure 4. Kanban helps agencies visualize and manage the Lean flow of epics.

Other than the first (Funnel) and last (Done) columns, all other columns have WIP limits. This visualization can help an agency avoid making commitments beyond its capacity to deliver, thus ensuring a smooth flow of epics from initiation to completion. This results in delivering the highest amount of value in the shortest sustainable lead time with quality, given the technology development dollars invested. Note that the transitions between each column in the Kanban process can also provide natural alignment points to existing governance stage gates until Lean practices have been applied to the government model as well.

Enable Variability and Learning by Rethinking Requirements

Shifting from projects to product-focused epics is more than a simple change in terminology. The language used to define an epic is fundamentally different. Instead of the traditional Øshall statementsØ that articulate detailed requirements up front, epics use *the language of intent*. This provides maximum flexibility to determine the best solution to meet the agency mission. Some elements of a solution must be fixed (for example, the protocols and frequencies used by the fighter jet to communicate to the military GPS network). But the best capabilities emerge when the variable elements of the solution evolve through innovation and sound engineering practices, such as model-based systems engineering and set-based design. In the broadest possible terms, agencies should define epics focusing on the problem to be solved or the functionality required. ARTs can then be provided all the freedom needed to rapidly prototype and deliver high-value solutions based on validated learning and rapid iterative development.

Moving Forward

The next key concept for adopting SAFe in Government is **Adopting Lean budgeting aligned to value streams**.

NEXT

Learn More

[1] The GAO Agile Assessment Guide. 2020. <https://www.gao.gov/assets/gao-20-590g.pdf>

[2] Ries, Eric. *The Lean Startup*. Crown Publishing, New York, 2011.

[3] Ries, Eric. *The Startup Way*. Crown Publishing, New York, 2017.

[4] *Exceeding Value*. The Standish Group International, Inc., 2014.

https://www.standishgroup.com/sample_research_files/Exceeding%20Value_Layout.pdf

Last update: 13 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California](#)
[Consumer Rights](#)

English

Scaled Agile Framework



The budget is not just a collection of numbers, but an expression of our values and aspirations.

—Jack Lew, former U.S. Secretary of the Treasury

Adopting Lean Budgeting Aligned

Adjust Cookie Settings

to Development Value Streams

This is article five in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

As complex as the annual budgeting exercise seems in most commercial enterprises, it pales in comparison to the processes governments use to appropriate, authorize, and apportion funding for hundreds of agencies and programs. For the leaders responsible for managing large government technology initiatives, only two things are almost certain:

The actual funding allocation will be different than the previous year.

The total will undoubtedly be different from the amount requested in the budgeting process.

Even worse than the uncertainties of funding, however, are the traditional budgeting procedures that force siloed, up-front precision estimates of time, cost, and scope for building specific point solutions. They lead to inaccurate estimates, blame-game exercises when the reality doesn't match the plan, and the stifling of innovation and adaptability.

As Lean management practices have been adapted for technology development, better budgeting and fiscal governance patterns that support innovation work have emerged. This article describes the shift from budgeting and funding government technology projects to planning and executing funds aligned to development value streams. (For an orientation to the topic, read the articles [Lean Budgets](#) and [Lean Budget Guardrails](#)).

Details

The previous article in this series, [Transitioning from Projects to a Lean Flow of Epics](#), established the foundation for shifting to Lean budgets. Evolving from planning work in projects to the continuous flow of working features of a solution in a development value stream is a critical pre-requisite for moving to a Lean budgeting model. After restructuring how work is organized and executed, the next step is to align how that work is funded using the same flow-based model. The following sections provide three recommendations to assist government agencies in adopting Lean budgets.

Align Budgets and Appropriated Funds to Development Value Streams

Transitioning from individual projects to a lean flow of epics in a development value stream sets the foundation to request funding in the annual budget based on the comprehensive needs of the development value stream versus individual programs and projects.

Figure 1 shows a typical government agency with traditional silos and branches organized by knowledge domains. Each branch budgets for its staff plus an extensive collection of supporting contracts.

Figure 1. Most government solutions cross multiple organizational branches and require support from numerous disconnected contracts.

As a result, the funding (and therefore control) for delivering a solution is spread

across disparate functions that are often locally sub-optimized. For example, it's common for an Infrastructure branch to default to historical data to budget cloud and communications contracts as an aggregate without understanding the solutions' current consumption and future needs using those resources.

Infrastructure may not be involved early and throughout the development of a new solution. When that solution requires capacity not included in the annual planning process, Infrastructure can be caught off-guard and then become a bottleneck, potentially delaying solution delivery. Similar delays can occur with testing, security, and other branches. It's also difficult to tie the consumption of these limited resources back to the appropriate solutions and value streams. In the U.S., this last challenge has prompted the adoption of Technology Business Management (TBM) to gain better visibility into how IT funds are spent [1, 2, 3].

Based on the principles of [Lean Budgets](#) and [Guardrails](#), Figure 2 offers a better solution. It assumes that the organization has identified end-to-end development value streams, for example, all of the financial systems in an agency, all of the solutions in each value stream (GL, AR, AP, Payroll, grants management, etc.), and the priorities for horizons of investing for each solution in the coming budget year.

Figure 2. Budgets are assembled based on the holistic needs of the development value stream and the solutions it contains.

Instead of each branch (Infrastructure, Security, Testing, etc.) submitting siloed departmental budgets, the leaders contribute their funding needs to Agile Portfolio Operations (or the equivalent office in large programs) to support each development value stream. Based on the vision and roadmap for each value stream, this group consolidates and negotiates priorities, along with forecasting the anticipated required funding levels. To build the funding requirements collaboratively for the coming year, trade-offs can be made using exercises such as [Participatory Budgeting](#) (also see [Lean Budgets](#)).

For example, a financial system's development value stream may have requests from Infrastructure to do a significant modernization initiative in the next fiscal year to move all remaining solutions to the [Cloud](#). However, the Finance department might raise a higher priority request to adapt all solutions to comply with the new tax laws so that payroll for employees and payments to suppliers will be correct when the law takes effect the following year. Given a limited projected total budget for the entire development value stream, the trade-off in participatory budgeting might be to defer the cloud migration for the least used systems until the following year, freeing up funds to build the new complex taxation rules into the applications. This demonstrates collaborative systems thinking across silos applied to the budgeting process for development value streams.

The net effect is that once the shift to lean budgets is complete, the agency CIO's annual budget requests are still reviewed and forwarded for consideration in the overall federal budget lifecycle process. However, now the budget will be organized by development value streams, not by individual branches or projects. The funds that flow down in the final appropriation and apportionment will also be subsequently allocated to the development value streams. It will then be up to the decentralized decision-making of the leaders in each development value stream to determine how the funds will be used throughout the year (a key to enabling agency agility).

Use Agile Metrics to Track Program Spend and Flow of Value

To ensure the government is getting the intended value for money spent, lean budgets also require different metrics for oversight and governance. Traditional development practices depend on models such as Earned Value Management to formulate final time and cost estimates based on planned versus actual progress

and expenditures. By contrast, Lean-Agile development uses real-time, objective evidence of progress based on working features of the system (whether new development, enhancements, or fixes) that are delivered every PI. Actual cost and capacity consumed per feature can be compared to planned cost and capacity to generate a predictability index that would be the most accurate indicator of overall program performance.

Figure 3. SAFe provides a wide array of metrics and management tools to monitor program performance.

Other metrics—such as feature progress charts, ART Kanban boards, burn-up charts, and continuous flow diagrams (Figure 3)—make program performance highly visible and transparent, enabling better fiduciary control. Agencies adopting TBM may consider the automated tools designed to support that framework to enhance further the visibility, management, and decision-making of technology expenditures. For a deeper look at metrics, see [Measure and Grow](#).

Exercise Fiscal Governance with Dynamic

Budgeting

Lean budgeting using SAFe provides two significant advantages over traditional budgeting. First, because each development value stream consists of long-lived ARTs that plan and forecast in cadence and synchronization, the visibility and predictability of spending improve significantly. Labor costs do not fluctuate wildly and can be efficiently allocated to individual features delivered by each PI. Giving fiduciaries a seat at the table in Lean Portfolio Management (LPM) activities and backlog refinement workshops also provides real-time visibility into how spending is aligned with agency priorities. Attending PI System Demos offers those fiduciaries exposure and thus confidence in the fit-for-use and fit-for-purpose of what was actually delivered in the PI. Financial executives often comment that once their organization shifted to Lean budgets, they had greater visibility and could execute their fiscal responsibilities more effectively.

The second advantage of Lean budgets is dynamic budgeting. Internal and external forces create more change and uncertainty for government agencies than ever before. All processes, including how technology budgets are developed and managed over time, must adapt to enable the organizational agility necessary for agencies to respond dynamically to meet mission requirements. Lean budgets using SAFe provide that flexibility.

In the example illustrated in Figure 4, an agency serving veterans has four major development value streams with solutions that provide the services shown in the graphic. Notice at the cadence-based boundary between PI 2 and PI 3 that something requiring a significant shift in agency priorities has occurred. Responding to this change will necessitate a shift in workforce assignments and associated funding.

An agency that has optimized for agility can respond quickly by reallocating teams or trains to the development value streams supporting the new priorities. Over time, as the theme of agility permeates everything from architectural decisions to contracting strategies, such a shift can be made by simply changing the backlog of work for one or more ARTs.

Figure 4. Lean budgets in SAFe enable agencies to shift budgets to respond to changing priorities.

In this example, one of the ARTs that had been working on features for systems in the Employment Assistance development value stream will now have features from the Education Assistance development value stream in their backlog starting in PI 3. Of course, this assumes that the technical skills of the ART are appropriate for the solutions in both value streams. If that's not the case, it may be necessary to change personnel on one or more trains. The potential delays, unplanned costs, and effects on team morale justify the cost and effort to design systems and contracts for maximum agility.

Address Ÿ Color of Money Ÿ Concerns

Among the common concerns regarding Lean budgets in a government context are the statutory restrictions frequently placed on appropriated funds. Funding authorizations can have strict limitations, such as the time window in which the

funds must be spent, the kinds of work activities allowed, and many more. The rules governing how every dollar (or euro, etc.) can be allocated are incredibly complex and beyond the expertise of most ART personnel.

Figure 5. Including fiduciary authorities in backlog refinement to classify work items resolves most funding classification concerns.

A simple, effective solution is to include fiduciary authorities who have the necessary funding classification expertise in each backlog refinement workshop. Being included in the refinement discussions lends an understanding of the work to be performed, allowing the fiduciary representative(s) to apply funding codes to each backlog item that then follows the item throughout its entire lifecycle. Concerns related to the funding of prioritized work in the backlog can be raised during refinement and then resolved before any work is performed. This is an instance where an automated Agile Lifecycle Management (ALM) tool is strongly recommended. All major tool providers support the metadata tagging of backlog items required for tracking and reporting by funding classification needed in the government context.

Moving Forward

The next key concept for adopting SAFe in Government is **Applying Lean Estimating and Forecasting in Cadence.**

NEXT

Learn More

[1] Leffingwell, Dean. *Agile and TBM: transforming the business of IT for the better.* Sep 11, 2018. <https://www.apptio.com/emerge/guests/agile-and-tbm-transforming-business-it-better>

[2] Accelerating the Mission: Recommendations for Optimizing Federal Technology Cost and Value in the Age of FITARA. <https://www.tbcmouncil.org/us-federal-it-cost-commission>

[3] Accelerating Business Value with SAFe and Technology Business Management (white paper). https://scaledagileframework.com/wp-content/uploads/2019/04/White_Paper_TBM_DIGITAL.pdf

Last update: 14 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)

[Watch and download
SAFe videos and
presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT &
TRADEMARKS

[FAQs on how to use
SAFe content and
trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE,
INC

CONTACT US

5400 Airport Blvd., Suite
300
Boulder, CO 80301 USA

BUSINESS
HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California
Consumer Rights](#)

English

Scaled Agile Framework



Our world is increasingly complex, often chaotic, and always fast-flowing. This makes forecasting something between tremendously difficult and actually impossible, with a strong shift toward the latter as timescales get longer.

[Adjust Cookie Settings](#)

~Andrew McAfee,

Author of *Machine, Platform, Crowd: Harnessing Our Digital Future*

Applying Lean Estimating and Forecasting in Cadence

This is article six in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

One of the most challenging transitions in adopting Lean-Agile in government is the shift to incremental flow-based models for cost estimation and schedule forecasting. Legacy practices grounded in traditional project management methods require comprehensive cost and schedule estimates up-front. Careful control mechanisms are put in place to detect any variance to the plan while the solution is being built. Any change then requires detailed justification, approvals, and change orders that can cause delays and cost overruns. To mitigate these change events and still make a profit, contractors have managed the risk this model incurs by padding their time and cost estimates, or capturing the shortfall in future change orders.

Lean-Agile estimation and forecasting practices recognize that technology development frequently involves unknowns that make precise estimates challenging. This is especially true when building solutions or capabilities that have never been built before, where the cone of uncertainty is very wide at the beginning. When the effort is smaller and/or the work is well-known, estimation has less risk and variability. It can be developed accurately from historical data. Lean-Agile also provides for agency adaptability to respond to changing conditions that necessitate adjustments to the vision, roadmap, and backlog.

This article describes SAFe Lean-Agile alternatives to traditional cost and schedule estimating practices. (Note: The previous articles in this series provide foundational concepts such as [Epics](#) and [Value Streams](#), so be sure to read them before continuing with the sections below.)

Details

Government leaders have a fiduciary responsibility to properly manage and account for the expenditure of taxpayer funds in the execution of agency missions. For technology development and operations that frequently require the use of external contractors, public sector leaders have the added responsibility of ensuring that the government receives quality solutions and services at fair prices.

Building, enhancing, fielding, and supporting technology-based capabilities typically begins with a government cost estimate and schedule of the initiative. In the U.S., the independent government cost estimate (IGCE) is the common process used to generate estimates for proposed programs to obtain approvals and funding before contracts can be awarded [1], [2], [3]. In virtually all countries, government program managers and contracting officers are responsible and accountable for ensuring technology capabilities are delivered within time, cost, and scope constraints. This estimating process often uses historical rates and information from previous, similar initiatives. But detailed, bottoms-up task breakdowns can also be used to justify the level of effort required.

SAFe recognizes that enabling agencies to plan and to manage ongoing PI execution in a Lean-Agile environment still requires estimating and forecasting. What's needed is an approach that:

- Balances fiduciary governance requirements with the ability to adjust estimates against macro-level constraints
- Accounts for both learning and for changing conditions

This is key to agency agility.

The sections that follow provide guidance on how to balance these concerns.

Estimating Using Epics

Some government programs are relatively easy to estimate and forecast. Take the example of a mature business system that requires only minor enhancements and bug fixes. For ongoing budget requests and contracting actions, cost estimates based on the historical quantity and skill levels needed, combined with the projected period of performance, can yield reasonably accurate projections.

However, the most impactful government programs are far more complex, involving the creation of new software and hardware solutions to enable entirely new capabilities. Examples range from re-platforming and modernizing mission-

critical software to systems that leverage cutting-edge technologies, such as artificial intelligence (AI), autonomous vehicles, and new-age composite materials. In other words, they haven't been built before, which means even historical data can be illusionary.

Detailed up-front designs and proposals are often used to estimate these programs. However, these can create false precision and unwarranted confidence in solutions that have many unknowns. History is littered with failed "well-specified" programs that were ultimately canceled without producing a working system after egregious time and cost overruns. A better way is needed.

The companion article, [Transitioning from project to a lean flow of epics](#), describes an epic as a large initiative of sufficient size and scope to warrant analysis, ROI assessment, a lean business case, and approval. The largest development programs—such as new weapons systems, or the wholesale replacement of a legacy financial system with an ERP—are likely to be a collection of related epics aggregated for governance purposes under a unifying program. Once a solution has been decomposed into one or more epics, Lean-Agile estimation can be used to develop cost estimates for each, and the program as a whole.

Because epics capture large complex initiatives that often have lots of uncertainty, a common approach for Agile estimation is to decompose the larger system or subsystem into smaller pieces of functionality called [Features](#) (Figure 1) that can be thought through individually. In SAFe, features (and their companion technical "Enablers") are what [Agile Release Trains](#) (ARTs)—the team-of-teams that build the system—deliver and aggregate into the final solution. Practitioners on the ART continuously improve their proficiency in estimating the level of effort required to deliver features in the common currency of story points. (Note that these are high-level, order-of-magnitude estimates for forecasting, not the more detailed estimates that ARTs will create during actual planning and execution).

Figure 1. Epics are decomposed into features and enablers for estimation and then aggregated back for epic sizing.

As long as government programs and contractors follow the Lean-Agile principle of long-lived teams-of-teams, the historical patterns of story points consumed to build features can be used to derive a cost-per-story-point metric for each ART.

Multiplied by the estimated story points per epic, this calculation is typically sufficient and reasonably accurate for such exercises as an ICGE or equivalent.

Program leaders still use historical data from similar features and epics as the best reference for estimating new epics (a current best practice even in traditional estimation). The primary difference between this process and the legacy approach is that estimates are based on the decomposition of the functionality of the *product*. This is not a detailed breakdown and estimation of the work tasks or techniques such as source lines of code (SLOC). If the work is entirely new and has many unknowns, prototypes and spikes may be needed as precursors to gain the learning needed to formulate reasonable estimates.

While this technique of decomposition to support estimation remains in use by many programs, additional cost estimation techniques describing estimating MVP costs, as well as forecasting implementation costs using T-shirt sizing based on historical data, can be found in the [Epic](#) article.

Forecasting Deliverables from the Backlog

Government fiduciaries not only need estimates of cost; they also need to be able to forecast when new capabilities can be delivered. To build Lean-Agile forecasts in SAFe, program leaders will require:

1. Long-lived Agile teams-of-teams (ARTs) with stable, dedicated practitioners

2. A prioritized backlog of estimated Epics (and the domain knowledge required)
3. An understanding of the historical velocity and available capacity of each ART that could potentially work on each epic for the next several PIs

With these three data points, leaders can formulate multiple “what if” scenarios to determine when each prioritized epic can be delivered.

Figure 2 provides a simple illustration of this process.

Figure 2. Forecasting Epic delivery dates using ART velocity and capacity allocation.

In this example, a government program has three ARTs (each ART represents 50-125 SAFe participants). Based on historical data of actual delivery capability, we know that ART 1 can deliver ~1200 story points per PI, ART 2 can deliver 1000, and ART 3 can deliver 650 (the differences are likely due to the varying number of personnel per ART). The highest priority epic has been estimated at 1000 story points. In the planning process, each ART must communicate how much of their total capacity per PI they can potentially dedicate to working on the top epic. If the skill sets in each ART are similar, there are many possibilities for how the work

could be divided and thus when the epic could be delivered. For example, if ART 2 has the skills to work on epic 1 and can devote 100% of their capacity, optimistically, that epic could be delivered in the first PI, or more realistically in PI 2 to account for issues and unknowns. To increase the probability of delivery in PI 1, features could also be distributed among multiple ARTs. However, if the skill sets in each ART are more domain-specific, then there is less flexibility, and planning will be needed to manage potential specialization bottlenecks.

Illustrating this what-if exercise can also expose where additional capacity is needed through addition or reallocation. Based on the agency's needs reflected in the backlog, adjustments were made to reallocate resources from ART 2 to the other ARTs in Figure 2, providing the necessary capacity for the highest priority epics. Making these types of adjustments in cadence, only at the PI boundaries wherever possible, is the best practice.

Additional techniques for estimating duration can be found in the [Epic](#) article.

Long-Range Forecasting and Roadmaps

Compared to more commercial software development initiatives, some government technology programs have very long-time horizons. This is especially true for cyber-physical systems such as fighter jets, satellite systems, and mission-critical applications. For this type of program, the normal three-PI (24-36 week) rolling-wave roadmap is insufficient. These large solutions require forecasting over many PIs—sometimes many years—to provide effective governance.

Figure 3 shows an example of a multi-year roadmap that forecasts when the component epics of an autonomous vehicle will be delivered, leading up to the initial operating capability (IOC) date near the end of the fourth year of the program.

Figure 3. Complex programs require a multiple year forecasting roadmap.

Figure 3. Complex programs require a multiple year forecasting roadmap.

The key to using this long-range planning tool while maintaining a Lean-Agile process model is to remember this artifact is a *forecast*, not a commitment. Over the life of the program, fact patterns and the dynamics of the environment will inevitably emerge, requiring an update to the forecast based on the best knowledge available. It's essential that government fiduciaries understand and embrace this mindset and avoid driving program leaders back into false precision estimating practices that have proven to have a high risk of failure.

Moving Forward

The next key concept for adopting SAFe in Government is **Modifying Acquisition Practices to Enable Lean-Agile Development and Operations**.

NEXT

Learn More

[1] Independent Government Cost Estimate (IGCE) Summary, U.S. Department of the Interior. <https://www.doi.gov/cloud/faq/igce>

[2] Independent Government Cost Estimate (IGCE) Handbook for Services Acquisition, Department of Defense.

https://www.acq.osd.mil/dpap/sa/Policies/docs/DoD_IGCE_for_SA_Handbook.pdf

[3] Independent Government Cost Estimate (IGCE) Template, National Oceanic and Atmospheric Administration (NOAA). <https://nitaac.nih.gov/resources/tools-and-templates/independent-government-cost-estimate-igce-template>

Last update: 9 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

TRAINING

[Linkedin](#)

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



*There is a way to do
it better & find it*

— Thomas A. Edison

Modifying Acquisition Practices to Enable Lean-Agile Development and Operations

[Adjust Cookie Settings](#)

This is article seven in the SAFe for Government (S4G) series. Click [here](#) to view the

S4G home page.

The most frequently cited barrier to Lean-Agile and DevOps transformation in government is the legacy acquisition process. Because agencies rarely have all of the personnel, experience, and material needed to create and operate technology-based capabilities, governments rely on industry partners to develop and support mission-enabling solutions. Regulations and guidelines that govern acquisition processes were written to protect citizen interests while creating a fair playing field for businesses competing for contracts.

However, these same guidelines are rooted in legacy acquisition processes that are often too rigid to meet today's demands for efficiency, innovation, and protection. Instead, they frequently lead to unreasonable delays in contract awards and Big-Up-Front Design (BUFD) processes that force a waterfall development method. Outcomes are often poor. This creates taxpayer dissatisfaction and distrust between contractors and government.

The knowledge pools of Lean-Agile development and DevOps offer an alternative approach. These methods clearly establish the need for a different mindset, as well as new values and principles to govern the new way of working. Trust and transparency among all parties are essential to building the best solution possible within time, cost, and scope constraints. Contractual language that requires big up-front design and full requirements defined early in the cone of uncertainty contradicts this approach.

This article describes how government agencies can expand acquisition practices to include alternative structures, terms, and conditions that enable a Lean-Agile development model while protecting the fiduciary interests of citizens.

Details

Government contracting officers are responsible for ensuring the proper and lawful use of taxpayer money to acquire products and services that enable agency missions. Indeed, in many countries contracting officers can be held *personally* liable for the misuse or mismanagement of government funds. Naturally, acquisition officials are risk-averse and cautious about making significant changes to the proven practices and contract language that have stood the test of time and legal challenges.

However, the terms and conditions needed for managing a traditional waterfall program compared to Lean-Agile are significantly different. In fact, expecting contractors to use Lean-Agile ways of working governed by waterfall contract

language and deliverables has proven to be a formula for failure. Figure 1 below illustrates just a few of the significant contrasts between historical acquisition practices that create anti-patterns for working in a Lean-Agile manner.

Figure 1. Comparison of the contracting practices used in waterfall and Lean-Agile programs.

The good news is that this is no longer new and uncharted territory in government agencies. Adoption of Lean-Agile and DevOps has grown to the point in the public sector where even the most conservative acquisition officials recognize that these are legitimate process models that require different contracting terms and conditions. Some governments are authoring new guidelines for soliciting, awarding, and managing contracts that support and enable Lean-Agile development (for example the U.S. Department of Defense Software Acquisition Pathway policy). Some even require it.

The sections that follow provide broad guidance for how government organizations can incorporate contracting best practices to support Lean-Agile, DevOps, and their instantiation in SAFe.

Embrace the Goals of an Agile Acquisition Model

First, the shift to Lean-Agile requires more than just changing terminology and activities. Iterative, flow-based development built on the foundations of Lean and Agile demands a shift in mindset, values, and principles, as well as a new set of practices. For example, one of the Òbig fourÓ values of the Agile Manifesto is Òcustomer collaboration over contract negotiation.Ó This reflects a vital shift in how both government and industry need to approach the contracting process. To enact this change, however, new values and principles MUST be reflected in the language of the contract.

The goals of a new set of acquisition guidelines to support Lean-Agile contracts should include the following:

- Optimize the economic value for all parties in the short- and long-term
- Provide adaptive responses to new requirements and faster value delivery
- Ensure complete transparency and objective evidence of program performance
- Enable a measured approach to investment that can vary over time and stop when sufficient value has been achieved
- Offer the supplier near-term confidence of funding and sufficient notice when funding winds up, winds down, or stops
- Motivate all parties to build the best solution possible within agreed-to economic boundaries
- Support Lean-Agile principles and practices

Unfortunately, existing boilerplate contract templates frequently promote the opposite of the principles described above. Government and contractors often have a low-trust, win-lose mindset. The language in legacy contracts frequently promotes waterfall processes, enforcing strict adherence to the plan with heavy variance reporting and change control. This limits flexibility and adaptability. In this environment, both sides tend to communicate information on a Ôneed to knowÕ vs. Ôneed to shareÕ basis.

Conversely, the language of a Lean-Agile contract should promote collaboration, transparency, adaptability, and a win-win mindset. Its aim should be to build the best solution possible within the established economic boundaries of time, cost, scope, and quality. Contract language needs to shift from project management, waterfall terminology (integrated master schedules, milestone review, phase gates, EVM, variance reporting, etc.) to Lean-Agile language (vision, roadmaps, backlogs, fast feedback, increments, objective evidence of working systems, etc.).

Develop Government Guidance for Agile Acquisitions

For most government agencies, the historical guidance for acquisition officials is deeply rooted in waterfall past practices. The long-term goal is to modify all policies, regulations, and ultimately statutes to accommodate Lean-Agile models. Change agents should focus on the easiest guidelines to modify first, which are typically agency-level policies. Such guidelines are usually generated on an agency-by-agency basis. Even if the governance office that owns the policy understands the need for an update to support the new way of working, getting the necessary changes approved can be a lengthy yet achievable process. The higher the level of government that owns the policy, the more difficult it will be to effect change. The longest and most challenging changes to make are those that require new laws to be passed by the legislative process. The good news is that in countries like the U.S. and the U.K., laws are starting to be modified to recognize Lean-Agile as a legitimate lifecycle model.

In the meantime, agencies should research the current state of agile contracting guidance and example contracts in their local context. Every country is at different stages in the agile journey. Some, such as the U.S., have made significant progress and are making statutory changes to enable (and in many cases encourage) a Lean-Agile approach. The last section in this article details some of these advances in the U.S. and the resources available to support agile acquisitions.

Agile Solicitation Language to SAFe Taxonomy and Processes

Agencies that have modified their acquisition processes to support Lean-Agile development may also have adopted SAFe as a Lean-Agile and DevOps implementation model. In these cases, acquisition officials can increase alignment with industry partners by creating the expectation in the contract that offerors should be prepared to follow SAFe practices if awarded the work. The graphic in Figure 2 illustrates critical elements to include in the solicitation to create a solid foundation for working in a SAFe environment.



Figure 2. Key elements to include in a solicitation for programs that will use SAFe.

In some cases, agencies can specify that SAFe is the lifecycle model contractors must follow. Other government entities preclude language that prescribes how the vendors will organize and manage their work. But even there, general agile language using a SAFe-like taxonomy can be a big help. When it's permissible, the more specific the solicitation can be, the greater the communication and alignment will be between the agency and its industry partners.

Access Additional Guidance

In the U.S. there has been substantial progress toward creating resources that provide agencies with the guidance, templates, and tools needed to author contracts that support Lean-Agile development. These are resources developed and published *by government agencies*, not by industry, for other government organizations to become more open to using these tools.

In the [SAFe for Government](#) article, there are two links in the opening section (Additional Resources and Videos) that include a wealth of resources, many published by government agencies, to support agile acquisitions. At present, most of these resources are designed for use by U.S. agencies, but as the adoption of SAFe grows internationally, this section will be updated with similar country-specific assets. Please check this page frequently and watch for updates in the SAFe Insider and blog page for announcements as new resources are added.

Moving Forward

The next key concept for adopting SAFe in Government is **Building in Quality and Compliance**.

NEXT

Learn More

[1] SAFe for Government Landing Page. <http://scaledagileframework.com/government/>

Last update: 10 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL
[Twitter](#)
[Linkedin](#)
[YouTube](#)

TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)
[Usage and Permissions](#)

INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



It takes less time to do things right than to explain why you did it wrong.

—Henry Wadsworth Longfellow

Building in Quality and Compliance

[Adjust Cookie Settings](#)

This is article eight in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

Local and national governments procure technology solutions for the welfare and security of their citizens. Indeed, governments define and procure some of the largest, most sophisticated, costliest, and most complex systems and solutions known to man. Many of these systems are high-assurance systems where the human and economic costs of failure are simply unacceptable (nuclear technologies, space stations and spacecraft, medical devices, intelligence gathering systems, etc.). And now, with the rise of modern cyber threats, security failures of many types of systems (citizen data, social and welfare systems) can be equally catastrophic.

To assure the safety and efficacy of these programs, governments historically mandated that stage-gated and sequential “waterfall development” processes be used to build these systems. Characteristics of this model include:

- Government fully specifies the system up front
- Suppliers build the solution to specifications
- The as-built system is passed to separate organizations for quality and compliance testing

Unfortunately, defects found late in the process often initiate massive redesign and rework (or worse, defects are not found, creating vulnerabilities to the system). Because of the rework, it might be months or years after a capability is initially developed before the solution can be fielded and the value received. In the interim, the mission needs often have changed so dramatically that by the time the solution is deployed, this expensive new technology is already out-of-date and no longer fit for its intended purpose. This often results in a massive waste of taxpayers’ money, subjects them to unnecessary risk, and denies them the benefits these systems were intended to provide.

The standard waterfall practice of deferring testing and quality activities to the end of a project is a significant contributing factor to these delays. Obviously, however, taking shortcuts to quality and compliance is not the solution. What is needed is a way to execute high assurance systems development in a way that verification, authorization, and accreditation activities are not pushed to the end of the development lifecycle. Any such alternative approach must enable faster delivery of such systems without decreasing the rigor or effectiveness of assurance outcomes.

This article describes the approach embodied in SAFe. While many of these concepts are already well documented in the Compliance article on the SAFe website, and further detailed in our white paper “Achieving Regulatory and

Industry Standards Compliance with the Scaled Agile Framework™ (SAFe™),¹ (downloadable [here](#)), this article describes quality and compliance guidance that is particularly applicable to the government context.

Details

No matter the development approach, government programs must integrate quality and compliance concerns into staffing, budgeting, and planning discussions. Technology initiatives have to satisfy numerous compliance standards for both the technical attributes of the solution and the processes used to build it. They might be based on international standards such as ANSI, ISO, CMMI, etc., or country-specific laws and the policies and regulations of particular government agencies.

Traditionally, most quality and compliance activities have taken place at the end of a project primarily because, as the top of Figure 1 illustrates, a working solution is not available for testing until the end. But the bottom of the Figure illustrates how an agile approach to building a system incrementally presents an opportunity to change how compliance work is conducted.

Figure 1. Agile development enables continuous quality and compliance checks

Figure 1. Agile development enables continuous quality and compliance checks

With SAFe, compliance activities can focus on smaller parts of the system as they are built. Two critical benefits explain how this improves the quality and compliance of the system:

First, compliance and quality officials have greater visibility and more opportunities to provide feedback when they are inspecting smaller parts of the system. (It's easier to find defects in 500 lines of code compared to 500,000 lines of code.)

Second, any quality and compliance issues discovered can be fixed more effectively early, *when the cost of change is small*. If not corrected early, other parts of the system will have been built on defective components, thereby creating new issues of their own, and exponentially compounding the effort required to fix the problem.

Clearly, agile development is a better approach. The remainder of this article

describes three essential practices that government organizations can use to build quality and compliance *into* the development process.

Integrate IV&V and Compliance Activities within Development

Many government agencies use independent verification and validation (IV&V) practices as part of the quality and compliance process. IV&V commonly occurs in a testing phase once the solution is completely developed. Instead of this work being performed separately following development, the Lean-Agile best practice is to integrate this work into the effort required to bring each Feature to completion, as illustrated in Figure 2.

Figure 2. Work to complete a Feature includes IV&V, audits, and compliance reviews

During backlog refinement and PI Planning, the Feature's size and scope include all of the work, *including* quality and compliance activities. Knowledge from those activities is then applied immediately as inputs to the next features to be

pulled from the backlog. Although frequent feature validation and review will not provide all the testing needed before releasing the solution, it can help make the final validation checks smoother and uneventful.

But making this shift is not a trivial effort. Government quality and compliance organizations and suppliers are often staffed and managed based on the phase-gated model. Conducting these activities in smaller batches, on smaller pieces of the solution every Program Increment (or even Iteration), requires changes to resource management and planning. If the organization doing this work is a third-party contractor, the change can also alter the terms and conditions of their contract with the government. Two strategies help mitigate these barriers to change.

First, many of these checks can be automated. Incorporating compliance tests into the automated build process ensures that the build does not pass unless the feature passes its compliance requirements. (And yes, even hardware-based systems can incorporate automation in the testing process.) This reduces manual verification and frees quality and compliance personnel to focus on higher-level activities.

Second, shifting to continuous IV&V can be incrementally tested and refined in pilot projects—before the approach is implemented across all agency programs.

Address Security Risks Early and Continuously

The second piece of guidance is much like the first, but focuses on the security aspects of the solution which require special consideration. Following DevSecOps best practices, government solution builders must include software and systems security practices into the development process and the system being developed. However, software and systems engineers may not always initially have the expertise needed to “build security in.” Mitigating strategies include:

- Incorporating DevSecOps specialists into the Agile Release Train (ART) as part of the Systems Team
- Adding a dedicated DevSecOps team to the ART
- Cross-training other practitioners in DevSecOps best practices
- Embedding DevSecOps experts into each Agile team

These experts can advise the team on best practices, assist with the design and implementation of DevSecOps infrastructure and architecture, help with the construction of automated security unit and systems tests, provide recommendations for the best static and dynamic security analysis techniques and help the teams advance their skills. They can also help the agency advance toward

a continuous Authority to Operate (ATO) model, such as the one being pioneered in the U.S. by the Department of Homeland Security.

Another emerging strategy is to build the solution on pre-approved Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) environments where applicable. The U.S. government provides FedRAMP certification for cloud platforms such as AWS GovCloud and Microsoft Azure Government an example of this approach. This allows the solution to inherit the security controls already built into these services. While that won't eliminate all security development activities, it can reduce them significantly and permit the teams to focus on security concerns that are application-specific. Teams may also benefit from leveraging published guidance from security governance organizations that provide continuous updates to reflect ongoing changes to the cyber threat environment. In the U.S. for example, these organizations include the National Institute of Standards and Technology (NIST) [1], and the National Cybersecurity & Communications Integration Center (NCCIC) US-CERT site [2]. Integrate Compliance Concerns into SAFe Artifacts and Events.

SAFe includes significant new guidance on DevSecOps practices in the [DevOps](#) article in the framework.

Finally, SAFe provides routine opportunities to incorporate quality and compliance concerns directly into the various aspects and events of the development process. Figure 3 illustrates just a few examples, each of which is described further below.

Figure 3. Quality and compliance concerns are integrated into routine activities.

Figure 3. Quality and compliance concerns are integrated into routine activities.

Kanban and backlog ☐ For proper estimation, prioritization, and dependency management, quality and compliance activities must be visible. Features should include stories that reflect the work needed to build-in quality and compliance as part of the definition of done (DoD).

Roadmap ☐ Even after shifting many of the compliance activities into developing the feature, quality and compliance milestones will continue to be critical points in the delivery timeline (for example, the final check flights for a new aircraft). These events should be reflected on the solution roadmap.

Solution Intent ☐ Quality and compliance artifacts, including test scripts and results, should be recorded in the Solution Intent repository. They should be used as inputs to the relentless improvement process.

PI Planning ☐ Whether they are incorporated directly into the composition of the ART, or interact as Shared Services, quality and compliance experts should be included in the PI Planning process. Compliance work should be reflected in a

team's plans and PI Objectives.

PI Demo D Teams should consider how quality and compliance can be demonstrated during System Demos as objective evidence that these requirements are being built into the solutions while they are being developed.

Inspect and Adapt D In addition to the demonstration portion of Inspect and Adapt (I&A), quantitative and qualitative metrics of quality and compliance should be presented during I&A. Any issues discovered, and the quality and compliance processes themselves, can become the focus of improvement efforts.

Moving Forward

The next key concept for adopting SAFe in Government is **Adapting governance practices to support agility and lean flow of value**.

NEXT

Learn More

[1] National Institute of Standards and Technology (NIST) Cybersecurity home page.
<https://www.nist.gov/topics/cybersecurity>

[2] National Cybersecurity and Communications Integration Center (NCCIC) US-CERT site. <https://www.us-cert.gov/>

Last update: 10 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



The speed of decision making is the essence of good governance.

ÑPiyush Goyal,
Minister of Railways
and Coal,
Government of India

Adapting Governance Practices to

Adjust Cookie Settings

Support Agility and Lean Flow of Value

This is article nine in the SAFe for Government (S4G) series. Click [here](#) to view the S4G home page.

Governments are some of the world's largest purchasers of technology and technological solutions intended to address some aspect of the health and welfare of the citizens they represent. They also have the mandate to spend taxpayer money wisely, to avoid waste, and to actively manage the risk of failure of a program or a deployed solution. **Governance** over the amount spent and the efficacy of the solutions they procure is mandatory.

For many reasons, government programs must contend with multiple layers of oversight throughout the development lifecycle. In some cases, the guidelines are internal to the agency that owns the program, but in many instances, programs must adhere to policies set by other agencies or by law. Traditionally, the phase-gate or waterfall method was used to implement oversight. It's not surprising, then, that the stakeholders responsible for governance are often concerned when changes are proposed to support the continuous value flow paradigm represented by Lean-Agile and SAFe.

This final installment of the [SAFe for Government](#) series addresses the challenge of how current governance practices perpetuate legacy waterfall processes and thereby do not necessarily accomplish their intended purpose of managing risk and spending. These legacy practices often produce the opposite results:

- Introduce unnecessary waste and delays
- Increase program risk
- Reduce overall economic outcomes for the taxpayer

The following sections provide recommendations on ensuring sound governance continues even after agencies adopt a continuous delivery lifecycle with SAFe.

Details

In this context, contractors and solution builders who simply want to deliver valuable capabilities as quickly as possible often view governance as a necessary evil. Much of this perception is driven by traditional oversight processes that are

seen as obsolete, redundant, and documentation-heavy, and only serve to delay the release of critical functionality. Interactions between program practitioners and “the inspectors” can be strained if not openly combative, with each side blaming the other for defects and delays.

To be fair, governance is *not* the natural enemy of the lean flow of value in technology development. In fact, governance plays a vital role in many high-assurance systems where the economic and human cost of failure is unacceptable. It is merely *a framework for decision-making and oversight to ensure that programs achieve the desired mission outcomes*. The dimensions of effective governance include:

Objectives → did the program deliver the intended capabilities

Constraints → did the program deliver within time, cost, scope, and quality guardrails

Compliance → did the program meet statutory, regulatory, and best practice guidelines

What’s needed is a re-evaluation of the *processes* used to exercise these oversight responsibilities while respecting the value governance provides to the ART and the various stakeholders of the resulting solution.

The following sections provide perspectives on how governance can evolve into a supportive and enabling process for technology programs using Lean-Agile, DevOps, and SAFe.

Evolving to Lean-Agile Governance

As with the other topics addressed in this series—such as budgeting, forecasting, contracting, etc.—the first step on the journey to adopting governance practices to support Lean-Agile is to recognize the contrasting patterns between traditional oversight approaches and the ways these responsibilities are carried out in a Lean-Agile environment. Table 1 summarizes the differences.

Table 1. Lean-Agile adoption requires a shift in traditional approaches to governance

Lean-Agile methods do not argue that accountability for program planning and execution within established boundaries is unnecessary. On the contrary, the values, principles, and practices promoted by SAFe are frequently viewed as more rigorous than previous approaches due to the focus on transparency and evaluation of progress through objective evidence (i.e., demonstration of working systems as the primary measure of progress and performance). Governance remains important with Lean-Agile. As seen in the right column, what has to change is *how* the oversight responsibilities are carried out.

The comparisons in Table 1 can be summarized by the need to adapt and recognize that technology programs can have many unknowns in the early stages of development. This often prohibits precise estimates at the detail level for requirements, design, tasks, etc. Traditional practices assume that all aspects of system development can be known up-front and that success can be achieved as long as variance from initial plans is tightly controlled. However, evidence from decades of failed projects has proven that an iterative and incremental model that permits discretion within broader time, cost, and scope boundaries is more

appropriate when the solution requires innovation and exploration into uncharted territory.

Some agencies in the U.S. government have recognized the need to adapt internal oversight regulations to account for the new way of working described by SAFe. They have modified their processes to include alternative models for providing governance using Lean-Agile programs. One of the most progressive case studies in the U.S. is the Department of Homeland Security (DHS). This agency has modified its guidance to assume Lean-Agile as the *default* planning and execution model for technology development. DHS's governance and oversight practices have evolved to align with this new way of working. (The governance document for DHS can be downloaded from the SAFe for Government [resources](#) page.)

The U.S. Government Accountability Office (GAO), the agency responsible for auditing federal programs, has also created a new guide for its auditors to use when evaluating programs that employ a Lean-Agile process model. [1]

Streamline Execution Reviews to PI Cadence

One of the success patterns we have observed in government programs using SAFe is streamlining the review process from a long list of assessments from different groups and committees to a consolidated review schedule that aligns with the PI cadence and events. One recommended model condenses all reviews into two types of milestones: the Incremental Capability Review (ICR) and the ART Backlog Review (PBR), as illustrated in Figure 1.

Figure 1. Milestone reviews consolidated and aligned to PI events

The ICR lines up with the Inspect & Adapt (I&A) and PI Planning events that occur in SAFe on the PI boundaries. I&A includes a comprehensive demonstration of system development progress during the previous PI. Since the features included in the PI for development are either working or not, the demo provides objective evidence of real progress. By providing key proof points that governance teams can closely watch, quality and compliance factors can also be demonstrated. I&A also includes a performance reporting item in the standard agenda and can include the required metrics to demonstrate the program is operating within various governance guidelines.

The PBR aligns to the frequent backlog refinement sessions (updating work remaining to be done) that occur in between the PI boundaries. This event is where program stakeholders make decisions regarding the scope and priority of the development backlog. Governance personnel focused on scope management can participate in each PBR for direct, real-time insight into what the program is delivering and the trade-off decisions that will inevitably need to be made as new information emerges.

Although many programmatic milestone reviews can be consolidated into the ICR and PBR, other milestone events will remain unchanged. For example, the test flight process for new aircraft can only be conducted once the plane is ready to fly.

However, with frequent and incremental checks throughout the development process, these final milestones will be more predictable, and the systems themselves will contain fewer defects.

Embrace the Inspectors

One common question is how to change governance models that are often entrenched in waterfall practices when the ownership of those policies resides outside the program's span of control. Clearly, such changes require patience and persistence. The current guidelines have evolved over many years. Adapting them to the new way of working will require a lot of consensus-building.

Change agents should start with a mindset of respecting and valuing the governance officials' role. Discussions of possible changes to policy and process should be built on a foundation of open, positive, and proactive relationships with the people with oversight responsibilities. Start from a position of "do no harm," acknowledging that any change to governance processes must still accomplish the original intent and enable governance teams to execute their responsibilities. New processes should be built collaboratively so that all parties have ownership of the new approach. In his book *Turn the Ship Around*, David Marquet describes this as "embrace the inspectors." [2] It recognizes that governance officials have intimate knowledge of the rules and have observed both good and bad implementation patterns across many programs. As a result, they represent a wealth of information and a voice that should be heard as new process models are formed.

Adopt Agile Earned Value Management (EVM) as Required

A common governance concern for government programs adopting Agile is the use of Earned Value Management (EVM) described in the Project Management Institute's (PMI) Project Management Body of Knowledge (PMBOK) to assess cost and schedule performance. While EVM is not a common practice globally for managing Agile development, some government agencies (particularly the Department of Defense in the U.S.) are currently mandated by law to use EVM on large programs. In these instances, whether EVM is useful or not is moot; program managers using SAFe in this context must meet EVM reporting requirements. Fortunately, government experts have already done much of the work to provide guidance for using EVM with Agile lifecycle models.

For programs using SAFe, the best practice to map backlog items to EVM equivalents at various levels is reflected in Figure 2 below.

Figure 2. SAFe backlog items can be mapped to equivalent constructs in EVM.

Figure 2. SAFe backlog items can be mapped to equivalent constructs in EVM

In this model, the top-level control account in EVM can contain and manage one or more Epics in SAFe. The Features in each epic map to planning packages when they represent future work on the [Roadmap](#) and evolve into work packages when detailed plans are created during PI Planning. Story details for each feature (scope, time, effort) provide the quantifiable backup data EVM requires. Since each feature planned represents X% of the total capacity of the ART for the PI, labor costs for the PI are known (material and other non-labor costs are a different matter). This makes the planned cost per feature easy to calculate. At the end of the PI, we also know the actual percentage of capacity of the train consumed to build each feature, so we have all the data needed to do the planned versus actual EVM measures.

While EVM ratios can be populated using this technique, they don't have the same utility in SAFe because the system is built in small batches and refactored or rebaselined at every PI boundary. Further, unlike a traditional project plan with

a static scope, Agile plans for change. After all, that's what creates agility. The net result is that the need for EVM in SAFe is not as great as in traditional waterfall projects. There are better, faster, and less complicated ways to get the same or more beneficial results when following Lean-Agile practices. But when it is mandated, SAFe practices can be adapted to provide the data required to meet EVM reporting requirements (for more information on mapping using EVM on Agile programs, refer to the Agile and Earned Value Management: A Program Manager's Desk Guide [3]).

Learn More

[1] GAO Agile Assessment Guide, Sep 2020.

<https://www.gao.gov/assets/710/709711.pdf>

[2] Marquet, L. David. *Turn the Ship Around!: A True Story of Turning Followers Into Leaders*. Portfolio, 2012.

[3] Agile and Earned Value Management: A Program Manager's Desk Guide.

https://www.acq.osd.mil/asda/ae/ada/ipm/docs/AAP%20Agile%20and%20EVM%20PM%20Desk%20Guide%20Update%20Approved%20for%20Nov%202020_FINAL.pdf

Last update: 15 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)

PARTNER

[Becoming a Partner](#)

CONTENT &
TRADEMARKS

SCALED AGILE,
INC

[Posters & Graphics](#)

[Watch and download](#)

[SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[FAQs on how to use](#)

[SAFe content and](#)

[trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

CONTACT US

5400 Airport Blvd., Suite
300

Boulder, CO 80301 USA

BUSINESS

HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



In a time of unprecedented technological advances and global change, Federal leaders, managers, and front-line employees must not only be aware of the changes to work, workforce, and workplace, but must also be able to identify weak signals, anticipate trends, and plan for

Adjust Cookie Settings

*the inevitable
changes.*

Ñ2022 Federal
Workforce Priorities
Report [1]

SAFe for Government

SAFe for Government is a set of success patterns that help public sector organizations achieve better solution development outcomes by implementing SAFe Lean-Agile values, mindset, principles, and practices.

The foundations of Lean and Agile thinking have led to higher success rates versus waterfall methods in software and systems development in the private sector. Government programs are starting to experience similar results using these same patterns. However, government agencies must address unique challenges in Lean-Agile transformations. The recommendations and best practices in SAFe for Government provide specific guidance to address these concerns.

The government-specific SAFe guidance is provided in the following series of nine articles:

- Building on a solid foundation of Lean-Agile values, principles, and practices
- Creating high-performing teams of teams of government and contractor personnel
- Aligning technology investments with agency strategy
- Transitioning from projects to a lean flow of epics
- Adopting Lean budgeting aligned to development value streams
- Applying Lean estimating and forecasting in cadence
- Modifying acquisition practices to enable Lean-Agile development and operations
- Building in quality and compliance
- Adapting governance practices to support agility and lean flow of value

The remainder of this summary article sets the context for why the principles and practices of Lean-Agile and SAFe are especially relevant for government agencies and provides a summary of each of the individual articles in the series.

(For more information on this topic, including links to pages containing a wealth of [additional resources](#) and [videos](#), check out our [Agile in US Government](#) page on [ScaledAgile.com](#).)

Details

Why SAFe for Government?

Lean-Agile and DevSecOps practices are gaining interest among the leaders responsible for the largest, most complex systems built for government worldwide. Much of that interest is driven by the internal and external forces changing how government agencies provide services to citizens, the government workforce, and warfighters. Governments are concerned with mission agility. While managing limited resources and funding is a concern, governments are not profit-driven but instead measure success as the value of impact that a solution provides. In this regard, governments resemble very large, complex non-profit organizations striving to run the business of operating as effectively and efficiently as possible. These imperatives drive:

- The need for business (mission) agility
- Impact of digital transformation
- The rise of social media and instant access to information about IT spending
- Increasing citizen expectations
- Technical debt and antiquated systems driving IT modernization initiatives
- Rapid changes in defense systems and the global cyber threat environment
- Fostering an Agile organization and growth mindset for the government workforce

Like for-profit organizations, the government is increasingly dependent upon technology. Yet, the traditional approaches to developing and sustaining solutions over the last 60 years have proven insufficient when developing modern technology-based capabilities. Agile practices have shown promise. However, the size and complexity of government systems, ranging from an unemployment benefits website for French citizens to large cyber-physical Sentinel missile systems, require more than team-level Agile and DevSecOps practices can provide.

Background of Agile Adoption in the U.S.

Federal Government

Interest in Lean-Agile methodologies for government technology development programs has increased exponentially in the U.S. since 2022. In July of that year, the U.S. General Accounting Office (GAO) issued a report recommending specific practices for Agile development, along with 14 unique challenges to Agile adoption in government. [2] That same year, the U.S. Office of Management and Budget (OMB) directed agencies to change their procurement practices from bloated long-

term projects to a more modular contracting approach aligned with an iterative development model. [3] Although these were positive signs that the decades-old commitment to waterfall processes was relenting, agencies were still slow to adopt a different way of working. Figure 1 shows a timeline of the significant events that have driven Lean-Agile adoption in the U.S. federal government.

Figure 1. Significant events that have driven Lean-Agile adoption in the U.S. government since 2010

The troubled launch of the U.S. Healthcare.gov website in the latter part of 2013 increased the interest in adopting Agile almost overnight. This site allows citizens to obtain health insurance as part of the Affordable Care Act (ACA). For weeks, the website's initial difficulties were highlighted in the national news, exposing many weaknesses in the traditional development practices common to government technology programs. [4] These events' public focus on government IT programs drove increased openness to adopt more modern development practices. In an analysis of Agile adoption in the U.S. Federal Government published by Deloitte in

2017, [5] the percentage of federal IT projects that report using Agile or iterative processes has grown dramatically since 2012, as shown in Figure 2.

Figure 2. Percentage of federal programs that report using an Agile or iterative development model

The increased interest in Lean, Agile, and DevOps accelerated when two new U.S. government agencies, 18F and U.S. Digital Service, were created to attract talent from industry to help bring modern, Silicon Valley-like practices to federal IT programs. The [Digital Services Playbook](#) and the [TechFAR Handbook](#) were two early resources they provided to help leaders in government programs understand how to modernize development practices and adjust the acquisition process to support Agile contracts. The number of additional government-authored resources on Agile adoption has grown significantly, as has the number of published success stories of federal programs getting better results after their transformation to Lean-Agile practices. One agency, the Department of Homeland Security, has made Agile the formal standard for software development. In September 2020, the Government

Accounting Office (GAO) released its draft of [The GAO Agile Assessment Guide: Best Practices for Agile Adoption and Implementation](#) for government programs using Agile practices. Congress continues to direct increased training in Agile as well as authorizing the modernization of acquisition practices to support Agile through legislation such as the annual National Defense Authorization Act (NDAA).

Agile Adoption in the Global Governments

Similar trends are being experienced in systems development for state and local governments and governments in countries across the globe. The UK government has been engaged for several years in an Agile transformation effort that spans hundreds of Agile teams across its many departments and agencies. PTMle Emploi (French employment agency), the Dutch Tax Authority, and the Australian Post (postal service) are examples of global government agencies using SAFe to guide their transformations to Lean-Agile at scale. In the case of PTMle Emploi, their transformation has resulted in better on-time delivery of employment benefits and increased satisfaction with the agency's services by both hiring businesses and job seekers.

Government as a Lean Enterprise

Increasingly, government agencies are challenged by the same forces of change driving their commercial counterparts to accelerate Lean-Agile transformations. The need for business (mission) agility, digital disruption, globalization, ever-increasing cyber threats, aging legacy systems, and increasing dependency on technology for business and mission success are just a few of the factors that are equally concerning for both government and industry.

The hallmark of a Lean Enterprise is the ability to deliver the best quality and value in the sustainably shortest lead time. SAFe provides guidance by describing the success patterns that help organizations achieve these competencies. The evidence provided by the many case studies of SAFe implementations in government agencies suggests that these competencies apply to public sector organizations as they do to commercial enterprises.

How SAFe Enables Lean-Agile and DevOps in Government

“In 10 months’ time, we have turned a failing effort into a success story for the warfighter, for our organizational culture, and for the taxpayer. We could not have executed a turnaround this fast without SAFe.”

Ñ Scott Keenan, JLVC Program Manager, Joint Staff (U.S. Department of Defense)

SAFe Adoption in Government is Growing

Many government technology programs are large and complex, involving hundreds (sometimes thousands) of practitioners. Solutions are built by multiple teams of teams that need to:

- Plan and work together
- Manage cross-team dependencies
- Integrate frequently
- Demonstrate working software and systems iteratively
- Share learning for relentless improvement

Often, these large solutions include small groups of government employees working closely with large numbers of contractor personnel. Multiple suppliers on different contracts may work on the same program across dispersed geographies. High assurance and compliance requirements, burdensome governance regulations, and abnormally long acquisition lead times further complicate government projects.

Most of these challenges are also common in large commercial organizations. Companies in the Global 1000 have used Lean-Agile to build core banking systems, satellites, farming combines, clinical and financial systems for health networks, and much more. For these organizations, SAFe has emerged as the leading Framework of practices for Lean-Agile at scale; the positive results have been documented in published case studies. Many government agencies have adopted SAFe as the process model for technology development for the same reasons as their commercial counterparts. Practitioners who have used SAFe in both contexts have reported that *there are far more similarities in development between industry and government than differences*. As shown in Figure 3, SAFe is now being used in hundreds of programs across a large number of government agencies.

Figure 3. Global government agencies with programs using SAFe as their development Framework

The Government has Unique Challenges Along the Path to Lean-Agile Adoption

Despite the increased momentum toward Lean-Agile and using SAFe, several barriers delay widespread adoption. The most frequently cited challenges include:

1. Poor implementations of Agile in the past created a reluctance to try again
2. Waterfall-centric governance and lifecycle policies that are not easily changed
3. An acquisition workforce that lacks experience with Agile contracts and Lean contracting practices
4. Project orientation, versus a continuous-flow-of-value mindset, is deeply ingrained in government culture
5. Long acquisition lifecycles create tremendous waste and delays in delivering value
6. The lack of a common enterprise Lean-Agile Framework leads to a limited synergy between programs

Although these problems look similar to those faced by commercial companies, the organizational context, culture, and governmental authorities in the public-sector environment are unique. Government acquisition processes and laws are intended to create a fair playing field among potential providers, but they can also create bureaucracy and delay unlike anything the private sector experiences. In addition, government agencies do not have the competitive market dynamic and profit motive that drives rapid change and innovation in a commercial environment. Instead, legislative bodies typically provide funding in politically-charged annual appropriations processes that move slowly. Even a government technology program's concept of "value" is often difficult to conceptualize and measure.

The Solution Ð Government-Specific Guidance for SAFe

Because factors in government technology development are specific to these environments, specialized guidance is required to help change agents as they lead transformations. The following links connect to articles addressing the most common challenges of adopting SAFe in government programs and the best practices that support transitioning to a Lean-Agile model.

Building on a solid foundation of Lean-Agile values, principles, and practices.

Waterfall ways of thinking and processes are deeply ingrained in government technology programs. Simply mimicking practices like the Team Sync and working from backlogs will not achieve agency agility. Government leaders and practitioners and their industry partners must understand how and why Lean-Agile fundamentally differs from past approaches to technology development.

Creating high-performing teams of teams of government and contractor personnel.

Lean-Agile development is a team sport. Teams on government programs often include a combination of government and contractor personnel. Too often, the relationship between these entities is more antagonistic than cooperative. This friction inhibits the development of high-performing teams and, ultimately, the rapid delivery of valuable quality solutions.

Aligning technology investments with agency strategy.

Government technology programs can be proposed, approved, and funded for various reasons. Technology investments are often made on the basis of assumed funding of past initiatives without the benefit of a periodic review of holistic portfolios of systems. The best use of limited development funds is achieved by ensuring that priorities are aligned and in harmony with the strategic imperatives of the agency.

Transitioning from projects to a lean flow of epics.

The start-stop nature of the Ôproject metaphorÕ for technology development leads to committing to point solutions too early in the process when there are too many unknowns and committing to delivering all planned features even if completing the highest priority features provides most of the value. Projects also encourage moving people to the work instead of flowing work to teams through single-piece flow backlogs. The Lean alternative is to organize large efforts into Epics and manage them in flow from a prioritized backlog built by long-lived teams of teams.

Adopting Lean budgeting aligned to development value streams.

Along with the shift from projects to a lean flow of Epics, there is a change in how to budget. Instead of funding individual pieces of work, budgets fund Ôthe factoryÕ that can build whatever the agency needs based on priorities. Since priorities can continuously shift and change, this approach avoids the wasted expense and delay of change orders or entirely new solicitations to shift investments from one

initiative to another, enabling agency agility.

Applying Lean estimating and forecasting in cadence. Lean-Agile practices recognize that traditional estimating and forecasting techniques often fail because there are too many unknowns when building entirely new systems. These traditional techniques also lock programs into the one right answer created up front and used to define contractual terms with multiple vendors. Lean estimating and forecasting techniques are lightweight and provide needed adaptability to constantly changing conditions while still providing critical reporting and accountability.

Modifying acquisition practices to enable Lean-Agile development and operations. Fewer barriers to Lean-Agile adoption in government are as daunting as the legacy acquisition process. Contracting officers depend on tried and true boilerplate language for each new acquisition firmly grounded in waterfall terms and conditions. Programs cannot be genuinely Agile if contractors are required to perform in ways that run counter to Lean-Agile values and principles. Transforming to a new way of development requires new templates for contracting officers that allow vendor partners also to be Lean-Agile.

Building in quality and compliance. The goal of Lean-Agile is to create a continuous and sustainable flow of value in technology solutions. Continuous flow is defeated if our governance processes, such as verification and validation, remain big batch activities performed only at the end of a project. Changing the way the development team works is only part of the answer. Every aspect of the technology value stream, including operations and governance, must also work in flow in smaller batches. This builds quality and compliance into the solution versus inspecting it as part of one big batch before release.

Adapting governance practices to support agility and lean flow of value.

Traditional governance standards for technology development are also deeply rooted in a waterfall model. They require programs to plan everything up front, commit to and budget for one right solution before the work begins, provide detailed project plans for unknown work, pass through arbitrary stage gates, and much more. Lean-Agile methods serve the original intent—*to provide sufficient oversight to ensure the delivery of mission-enabling capabilities within a reasonable time and cost—but through the use of alternative processes and artifacts that support a continuous flow of value.*

NOTE: These recommendations for Lean-Agile adoption in government do not require a different version of SAFe or suggest modifying SAFe terms and practices to fit government protocols. Experienced *practitioners in government services have reported that they achieve the best results when the SAFe model and*

terminology are used without modification. Using the terminology native to SAFe allows program personnel to take advantage of the classes, articles, books, forums, and other information sources necessary to implement SAFe practices successfully. The articles behind each of the practices above explain the specific patterns used by government programs to overcome the most common concerns when transitioning a federal IT initiative to SAFe.

Summary

Government adoption of Lean-Agile has accelerated to the point where most programs (80%+ in the U.S. according to a Deloitte study) are using some form of Agile or iterative development. However, Agile practices are often limited to development teams and do not address the program and portfolio challenges of strategy alignment, budgeting, project-centric planning, acquisitions, governance, compliance, and more. Agencies also lack a common language and set of enterprise-wide practices to create synergies between programs and practitioners. With this SAFe for Government guidance, supported by the corresponding SAFe for Government course, agency leaders will have the tools to overcome common barriers to SAFe adoption and Lean-Agile practices, enabling better results.

Learn More

[1] 2022 Federal Workforce Priorities Report. <https://www.opm.gov/policy-data-oversight/human-capital-management/federal-workforce-priorities-report/2022-federal-workforce-priorities-report.pdf>

[2] *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods.* General Accounting Office (GAO), July 2012.
<https://www.gao.gov/products/GAO-12-681>

[3] *Contracting Guidance to Support Modular IT Development.* Office of Management and Budget (OMB), June 2012.
<https://obamawhitehouse.archives.gov/blog/2012/06/14/greater-accountability-and-faster-delivery-through-modular-contracting>

[4] Brill, Steven. *Code Red: Inside the nightmare launch of Healthcare.gov and the team that figured out how to fix it.* Time, March 10, 2014.
<http://content.time.com/time/subscriber/article/0,33009,2166770-2,00.html>

[5] Viechnicki, Peter, and Mahesh Keikar. *Agile by the Numbers: A data analysis of*

- Agile development in the US federal government.* May 5, 2017.
<https://www2.deloitte.com/insights/us/en/industry/public-sector/agile-in-government-by-the-numbers.html>
- [6] Pawlikowski, Ellen. *USAFO's Pawlikowski: DoD Use of Agile Software Development Critical*. <https://youtu.be/nQUplJVjql>
- [7] *The Digital Services Playbook*. US Digital Service. <https://playbook.cio.gov/>
- [8] *The TechFAR*. US Digital Service. <https://techfarhub.cio.gov/handbook/>
- [9] *The TechFAR Hub*. US Digital Service. <https://techfarhub.cio.gov/>
- [10] *Modular Contracting*. 18F. <https://modularcontracting.18f.gov/>
- [11] *Digital IT Acquisition Professional Training*. Federal Acquisition Institute.
<https://techfarhub.usds.gov/get-started/ditap/>
- [12] *Agile Acquisitions 101*. Federal Acquisition Institute. <https://www.fai.gov/media-library/item/agile-acquisitions-101>
- [13] Eggers, William D. *Delivering on Digital: The Innovators and Technologies that are Transforming Government*. Deloitte University Press, New York, 2016.
- [14] *The GAO Agile Assessment Guide: Best Practices for Agile Adoption and Implementation*. <https://www.gao.gov/products/GAO-20-590G>
- [15] *Science & Tech Spotlight: Agile Software Development*. General Accounting Office (GAO), Sept 2020. <https://www.gao.gov/products/gao-20-713sp>
- [16] William D. Eggers, Allan Mills, Hans Verheggen and Carsten Joergensen. *Government Trends 2021- Global Transformation Trends in the public sector*. March 2021.<https://www2.deloitte.com/us/en/insights/industry/public-sector/government-trends/2021/agile-at-scale-in-government.html>
- [17] DoD Software Modernization Strategy. U.S. Department of Defense. Feb 2022.
<https://www.defense.gov/News Releases/Release/Article/2924187/dod-software-modernization-strategy-approved/>

Last update: 15 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Specifically, you can take the time to develop and bring to the table an outside-in, market-centric perspective that is so compelling and so well informed that it can counterbalance the inside-out company-centric orientation of last year's operating plan.

Adjust Cookie Settings

Agile Product Delivery

Agile Product Delivery is one of the seven core competencies of SAFe, which is essential to achieving [Business Agility](#). The [Measure and Grow](#) article provides a self-assessment for each competency, including APD, to evaluate a team's proficiency and identify improvement opportunities.

Why Agile Product Delivery?

Achieving business agility requires [Agile Teams](#) and [Agile Release Trains \(ARTs\)](#) to increase their ability to deliver innovative products and services rapidly. This capability requires balancing the focus on execution and customers, ensuring the creation of the *right solutions*, for the *right customers*, at the *right time*.

Figure 1 illustrates the three dimensions of APD:

1. Customer Centricity and Design Thinking
2. Develop on Cadence, Release on Demand
3. DevOps and the Continuous Delivery Pipeline

The mutually supportive capabilities of the APD competency create opportunities for sustained market and service leadership.

Figure 1. Three Dimensions of APD

Figure 1. Three Dimensions of APD

The following sections describe each dimension of APD.

Customer Centricity and Design Thinking

Customer centricity and design thinking comprise the first dimension of APD. This mindset and way of doing business put the customer first, at the enterprise's core, to provide positive customer experiences and build long-term relationships.

Customer Centricity

Customer Centricity is a mindset and way of doing business that focuses on creating positive user experiences and customer engagement with the organization's products and services. It puts the customer at the center of every decision, deeply considering the effect on its end users [1]. This mindset motivates long-term customer relationships, enabling more customer value, often in unexpected ways

[2]. This dimension of APD encourages Agile Teams to:

Focus on the customer **D** Employ user and market research, including developing personas, to align and focus the organization on specific, targeted user segments.

Understand the customer's needs **D** Invest time in identifying and building solutions that address these needs.

Think and feel like the customer **D** Apply empathy and strive to see the world from their customers' point of view.

Build whole product solutions **D** Design whole product solutions for the user's needs, ensuring that the initial and long-term user experiences are optimal and evolve as needed.

Create customer lifetime value **D** Move beyond a transactional mentality and focus on the overall customer relationship over the solution's life. [2].

Customer-centric businesses create greater profits, employee engagement, and customer satisfaction. Customer-centric governments and nonprofits create resilience, sustainability, and the alignment needed to fulfill their mission.

[Product Management](#) is responsible for coordinating and bringing new solutions to market while ensuring the ongoing success of existing products.

Design Thinking

[Design Thinking](#) is integral to *customer centricity*. It's an iterative development process that ensures solutions are desired by customers and users while also ensuring the solution is feasible, economically viable, and sustainable throughout its lifecycle.

It has two main activities that result in a sustainable solution:

Understanding the problem **D** The problem space is where designers explore the problem, including its complex nature, and get a clear definition of the problem, gaining insight into the requirements and benefits of a desirable solution

Designing the right solution **D** The solution space is where ideas are generated, visualized, and prototypes are developed and tested.

Figure 2 shows the core processes of design thinking, illustrated as a *double diamond*. This process focuses on thoroughly exploring the problem space before creating solutions.

Applying design thinking during development ensures the solution is *desirable*, *viable*, and *feasible*. At the same time, understanding and managing solution economics results in a *sustainable* product or service.

Figure 2. Design thinking process and activities

Figure 2. Design thinking process and activities

Understanding the problem typically includes the following two activities:

Discover ⚡ Seeks to understand the problem by engaging with users and market research to identify unmet needs.

Define ⚡ Analyzes the discover phase data using convergent techniques to generate insights into the specific problems and unmet needs.

After exploration, the organization has the inputs to begin designing a solution, which often involves the following activities:

Develop ⚡ Applies customer journey maps and story mapping to quickly design potential, cost-effective solutions.

Deliver ⚡ Produces various artifacts that are suitable for creating the solution. These solutions often start as prototypes with continuous delivery from the ARTs.

Figure 2 also illustrates how divergent and convergent thinking are applied to exploring ideas, working towards goals, and addressing challenges. Both are

necessary, and together they lead to unique solutions for challenges that require exploration and creativity.

Lean UX

In SAFe, [Lean UX](#) extends the traditional user experience design process beyond merely executing design elements and anticipating how users might interact with a system. Instead, it encourages a far more comprehensive view of why a [Feature](#) exists, the functionality required to implement it, and a *hypothesis* for its intended benefits. Leading indicators and getting immediate feedback from customers and end-users help determine if the system meets customer needs and business objectives. Lean UX provides a closed-loop method for defining, hypothesizing, building, measuring value, and learning.

In Lean UX, the designer's role evolves more toward design facilitation and taking on a new set of responsibilities. Besides Lean Startup, Lean UX has two other foundations: Design thinking and Agile development. Design thinking helps widen the scope of user experience work beyond mere interfaces and artifacts. It looks at the whole system and applies design tools to broader customer problems, relying heavily on collaboration, iterative approaches, and empathy as its core to problem-solving. [3]

Develop on Cadence, Release on Demand

Figure 3 illustrates the concept of developing on cadence and releasing on demand. It separates the concerns of *developing solutions* and *releasing* value, ensuring customers can get what they need when desired, which improves business agility.

Figure 3. Developing on cadence enables value to be released on demand

Figure 3. Developing on cadence enables value to be released on demand

Why Develop on Cadence?

In a flow-based system, establishing routine development activities on a fast, synchronized PI cadence—[a regular predictive rhythm of team and ART events](#)—is a proven strategy to manage the *inherent variability* in product development. The following activities support this cadence:

ART events □ The ART has several important cadence-based events: [PI Planning](#), [System Demos](#), and [Inspect and Adapt](#). The PO and Coach Sync events are held throughout the PI to help eliminate impediments, remove bottlenecks, and communicate adjustments needed by the teams.

Agile Team events □ PIs are divided into iterations, which help align Agile Teams and enable faster response to change. Team cadence-based events further support teams: [Iteration Planning](#), Team Syncs (usually held daily), [Iteration Review](#), and [Iteration Retrospective](#).

Simply put, teams use a process optimized for highly variable knowledge work, providing a reliable series of events and activities on a regular, predictable schedule.

Why Release on Demand?

[Releasing on demand](#) provides a significant strategic advantage by making value available when customers, the market, and the business need it. In collaboration with stakeholders, Product Management determines when a release should happen, what elements should be released, and who should receive it.

Some products serve market segments that offer new functionality as soon as it's available. While others may have distinct market rhythms that govern optimal release windows, as described in the [Roadmap](#) article.

Figure 4 illustrates the RoD processes by which new functionality is deployed into production and *released* incrementally or immediately to customers based on user or market demand.

Figure 4. The four activities of Release on Demand

Figure 4. The four activities of Release on Demand

Release describes the practices necessary to deliver the solution to end users, all at once or incrementally

Stabilize and Operate describes the practices needed to make sure the solution is

working well from a functional and non-functional perspective

Measure describes the practices to quantify if the newly-released functionality provides the intended value

Learn describes the practices needed to decide what should be done with the information gathered and prepare for the next learning loop through the CDP

Building and maintaining a [Continuous Delivery Pipeline](#) (CDP) allows each ART to define, build, validate, and release new functionality to meet their PI objectives.

DevOps and the Continuous Delivery Pipeline

[DevOps](#) and the Continuous Delivery Pipeline lay the foundation that enables releasing value, in whole or part, at any time to meet demand.

While releasing on demand is a goal of the CDP, gaining the competency to reliably and skillfully release value whenever desired is hard work. It involves embracing the DevOps mindset and culture and creating an increasingly automated pipeline.

Each ART builds and maintains (or shares) a CDP with the assets and technologies needed to deliver solutions as independently as possible. The first three aspects of the pipeline, [Continuous Exploration](#), [Continuous Integration](#), and [Continuous Deployment](#), support the delivery of new functionality, as illustrated in Figure 5.

Figure 5. The Continuous Delivery Pipeline

Figure 5. The Continuous Delivery Pipeline

Continuous Exploration promotes innovation and aligns with what should be built.

Design Thinking continually explores customer and market needs, defining a [Vision](#) and Roadmap.

Continuous Integration builds quality into the development process by continuously integrating many Agile Teams' work.

Continuous Deployment represents the processes associated with migrating solutions from staging to production.

As noted earlier, Release on Demand (Figure 4) is the ability to make value available to customers all at once or in an ad hoc fashion based on market and business needs.

Embracing DevOps Mindset, Culture, and Practices

High-performing organizations use DevOps to dramatically outperform competitors by delivering and supporting their products and services to respond to customer demands faster.

Figure 6 illustrates that Dev is frequently in a *fast-forward* mode, trying to keep

pace with the constant demand for changes and innovation. At the same time, Ops often presses **pause** on changes because they are accountable for production stability and resiliency.

DevOps aligns efforts across development, operations, and other business functions to achieve an optimal balance of speed and stability.

Figure 6. DevOps fosters collaboration across all functions

Figure 6. DevOps fosters collaboration across all functions

Ultimately, DevOps is a mindset, a culture, and a set of technical practices that provides solution elements to the customer without handoffs or too much external production and operations support. As illustrated in Figure 7, SAFe's approach to DevOps is grounded in five concepts: Culture, Automation, Lean Flow, Measurement, and Recovery ([CALMR](#)), briefly described below.

Figure 7. SAFe's CALMR approach to DevOps

Figure 7. SAFe's CALMR approach to DevOps

Culture □ A culture of shared responsibility is needed for fast value delivery across the entire Value Stream. All relevant departments help create value, including development, testing, security, compliance, operations, architecture, and more.

Automation □ Automation is used to reduce or eliminate human intervention from the CDP to decrease errors and reduce the overall cycle time of the release process.

Lean flow □ Fosters limiting work in process (WIP), smaller batches, and reducing queue lengths. In other words, they [Make Value Flow without Interruptions \(Principle #6\)](#) and enable faster customer feedback.

Measurement □ Supports learning and continuous improvement by understanding and measuring the flow of value through the pipeline.

Recovery □ Builds systems that allow fast fixes of production issues such as automatic rollback and “fix forward” capabilities, immutable infrastructure, and more.

Cloud Computing is a Key Enabler of DevOps

The ever-expanding universe of cloud capabilities has fundamentally changed how digitally enabled solutions are built, deployed, and maintained. Cloud computing has been one of the most disruptive drivers for changing the delivery model for enterprise IT since its inception [1]. Not surprisingly, the primary reason to move to the cloud is to increase product development speed and agility [2].

The cloud is everywhere, and it fuels digital business and enables DevOps and a more efficient CDP. SAFe enterprises can harness the power and ubiquity of the

cloud to increase agility in all areas of the organization.

Team and ART Flow

Since SAFe is a flow-based system, any interruptions to flow must be resolved quickly to enable continuous value delivery. SAFe provides six articles to help fix impediments to flow: Principle #6 – Make value flow without interruptions, [Value Stream Management](#), [Team Flow](#), [ART Flow](#), [Solution Train Flow](#), and [Portfolio Flow](#). Each article defines a set of eight flow accelerators that help identify, fix, optimize, and debug issues to achieve a continuous value flow.

The ART and Team flow guidance apply directly to the APD competency:

ART Flow – This represents a state where an ART delivers a continuous flow of value to the customer. It describes how teams-of-agile teams (ARTs) work with their stakeholders to get closer to customers and build CDPs. The CDP accelerates the delivery of products and services.

Team Flow – This represents the state in which *Agile Teams* deliver a continuous flow of customer value. The SAFe [Team and Technical Agility](#) (TTA) competency offers practices for creating effective cross-functional Agile Teams and ARTs. It fosters applying built-in quality practices and collaborating with extended stakeholders to deliver solutions faster.

Summary

Businesses need to balance their execution focus with a customer focus to help ensure that they are creating the right solutions, for the right customers, at the right time. APD is grounded in customer-centricity, design thinking, and Lean UX putting the customer at the center of every decision. It applies design thinking to ensure the solution is desirable, feasible, viable, and sustainable.

Developing on cadence helps manage the variability inherent in product development. Release on demand separates the release and development cadences to ensure customers can get what they need when needed. DevOps and the CDP create the foundation that enables releasing value, in whole or part, at any time to meet customer and market demand. APD enhances business agility, offering superior outcomes for the enterprise and its customers.

Learn More

[1] Norman, Don. *The Design of Everyday Things*. Basic Books, 2013.

[2] Osterwalder, Alexander, Yves Pigneur, Gregory Bernarda, and Alan Smith. *Value Proposition Design: How to Create Products and Services Customers Want*. Wiley, 2014.

[3] Gothelf, Jeff, and Josh Seiden. *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media, 2016.

[4] Lean UX vs. Design Thinking. <https://uxplanet.org/lean-ux-versus-design-thinking-3f9ebb8aef59>

Last updated: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

Scaled Agile Framework



The productivity of software delivery at enterprise organizations falls woefully behind that of the tech giants, and the digital transformations that should be turning the tide are failing to deliver business results.

ÑMik Kersten,
*Project to Product:
How to Survive and Thrive in the Age of*

Adjust Cookie Settings

Business Agility

Business Agility is the ability to compete and thrive in the digital age by quickly responding to market changes and emerging opportunities with innovative, digitally-enabled business solutions.

Everything moves fast in the digital age. Customer desires, competitive threats, technology choices, business expectations, revenue opportunities, and workforce demands now happen at *blistering speeds*.

Today, achieving customer delight at the speed of the market requires validating innovations with customers and then Ôpivoting without mercy or guiltÕ when the hypothesis needs to change. Moreover, significant technological advances are unlocking new ways to create this value. For example, [AI](#), [Big Data](#), [Cloud](#), and [DevOps](#) enable enterprises to expand their product lines, modernize their existing offerings, scale to mass markets, make better fact and insight-based decisions, and streamline solution development.

Competing in the Age of Software

In her book *Technological Revolutions and Financial Capital*, Carlota Perez [2] explains the evolution of business, society, and financial cycles based on her analysis of five significant technological revolutions over the past three centuries. Her research begins with the *Industrial Revolution*, leading to the *Age of Steam and Railways*, the *Age of Steel and Heavy Engineering*, and the present *Age of Software and Digital*, as illustrated in Figure 1.

Perez concludes that these revolutions lead to massive social change, market disruption, and an all-new economic order. Indeed, these are *world-shaking* disruptions that typically occur once in a generation.

Figure 1. Technological revolutions change society

Figure 1. Technological revolutions change society

We are in the midst of one of those ages now, the deployment period of the **age of software and digital**. This period is when every business is a software business. Put simply, competing in this age requires large-scale software and system development capability that enables true business agility.

Why Organizations Struggle to Achieve Business Agility



The organizations we created in the 20th century were designed much more for reliability and efficiency than for agility and speed.

Ñ John P. Kotter [3].

Most leaders in traditional organizations are aware of the digital disruption threat, yet many fail to transition to take their place in the next economy. The question is, why?

Organizations Start as a Fast Adaptive Network

As an organizational researcher and author, John Kotter illustrates in his book, *Accelerate: Building Strategic Agility for a Faster-Moving World* [4], successful enterprises don't start as large and cumbersome. Instead, they typically began as a fast-moving, adaptive network of motivated individuals focused on responding to the customer and the new business opportunity. Roles and reporting relationships are fluid, and people collaborate organically to identify customer needs, explore potential solutions, and deliver value in any way they can. In other words, it's an adaptive entrepreneurial network of people working to leverage an opportunity (Figure 2).

Figure 2. New enterprises operate as networks focused on the customer and new business opportunities

Hierarchy Forms, then Grows and Grows

As the enterprise succeeds, it naturally wants to expand on its success and grow. This growth means that individual responsibilities must become clearer. As a result, the enterprise hires specialists to add expertise, creating new functional areas.

Policies and procedures ensure legal adherence and compliance, driving repeatable, cost-efficient operations. The business starts to organize functionally to scale, causing silos to form. Meanwhile, operating in parallel, the network continues to seek new opportunities to deliver value (Figure 3).

Achieving larger economies of scale requires the hierarchy to grow. And it grows until it conflicts with the entrepreneurial network.

Figure 3. Growing hierarchy running in parallel with an entrepreneurial network

Figure 3. Growing hierarchy running in parallel with an entrepreneurial network

The Hierarchy and Adaptive Network Collide

Eventually, the hierarchy collides with the faster-moving, more adaptive network. The result? The adaptive network gets crushed. The focus on the customer is often one of the main casualties (Figure 4.)

Figure 4. Entrepreneurial network collides with a growing hierarchy

Figure 4. Entrepreneurial network collides with a growing hierarchy

Without the entrepreneurial network, the organization lacks the agility to respond when the customer needs shift dramatically or when a disruptive technology or competitor emerges. An urgent crisis erupts, and the company's survival is now at stake.

However, the organizational hierarchies built over the last fifty years have provided time-tested structures, practices, and policies. They support the recruiting, retention, and growth of thousands of employees across the globe. Simply put, they are still needed.

In addressing this dilemma, Kotter points out, *“The solution is not to trash what we know and start over but instead to reintroduce a more agile, network-like structure”* that operates in concert with the hierarchy to create what he calls a “dual operating system.” This system, illustrated in Figure 5 and described in the following section, allows companies to capitalize on rapid-fire strategic challenges and retain their stability [3].

The Solution: SAFe is the Second Operating System

The existing hierarchy, people, and management still have a purpose and largely remain in place. However, SAFe creates a second *virtual* operating system organized around [Development Value Streams](#) instead of functional silos (or departments) to form the entrepreneurial network.

Each development value stream creates one or more [Agile Release Trains \(ARTs\)](#) with a shared business and technology mission. Each ART plans, commits, develops, and deploys together. They are an integral part of the entrepreneurial network that develops innovative products ([Solutions](#) and services).

Although the management reporting structure in the hierarchy may remain largely the same, the teams in an ART are self-organizing and self-managing, and they no longer need daily task direction. ARTs are virtual organizations with the people needed to define, deliver, and operate the solution. This new virtual organization breaks down the traditional functional silos that inhibit flow and innovation.

By organizing the second operating system around value streams instead of departments, SAFe offers a way for organizations to focus on customers, products, innovation, and growth in harmony with their existing hierarchical structure.

Moreover, this operating system is *flexible*. It is built on time-tested Lean-Agile SAFe practices. It can organize and *quickly reorganize* without completely disrupting the existing hierarchy, as illustrated in Figure 5. That's what business agility demands.

Figure 5. SAFe as the second organizational operating system

Responding to market changes and emerging opportunities is vital to surviving in the digital age, where disruption is the norm rather than the exception.

Technological advances have drastically shifted the dynamics of the competitive game by opening up various ways to win in the market. The ability to respond quickly with innovative business solutions—what we call Business Agility—is the deciding factor between success and failure. Therefore, enabling business agility is a mission-critical goal for every organizational leader.

An organization must first understand and then apply the SAFe® Business Agility Value Stream to deliver value faster to keep today’s customers and win new ones. The SAFe *Business Agility Value Stream (BAVS)* helps organizations visualize the steps and implement the SAFe core competencies needed to move from identifying an opportunity to delivering customer value in the shortest possible time. The following sections describe the BAVS in greater detail.

Core Competencies of Business Agility

Achieving business agility requires this dual operating system and a significant degree of expertise across SAFe’s seven core competencies, as illustrated in Figure 10. Fortunately, they are part of SAFe, and as organizations adopt the Framework, they acquire the competencies incrementally over time. While each competency can deliver value by itself, true business agility is only present when the enterprise achieves significant mastery of all. It’s a tall order, but the path is clear.

Figure 6. SAFe Core Competencies

Figure 6. SAFe Core Competencies

Each competency is described in an article of that same name. You can easily navigate each competency from the SAFe Overview tab or the SAFe home page.

Why the Business Agility Value Stream (BAVS)?

Digital transformation is advancing in almost all business processes. Technologies such as AI, big data, and cloud computing are unlocking possibilities for creating new customer value. New business opportunities emerge more frequently, and many have the potential to disrupt market incumbents. Companies that continuously leverage these technologies will acquire more customers and improve value to existing customers. Ultimately, they will dominate their markets. For example:

Customers who can seamlessly fulfill their primary banking needs with an easy-to-use mobile app will remain loyal to that bank.

Health organizations that pioneer virtual urgent care at home will earn the gratitude and loyalty of all those who need their help.

A driver whose vehicle gets smarter and safer every day is more likely to remain loyal to that automotive manufacturer.

Traditional Development Won't Get You There

In this new reality, competitiveness equates to rapidly delivering digitally enabled solutions. As Figure 7 illustrates, responding to a traditional, phase-gate development process may mean missing the opportunity.

Figure 7. The traditional approach to leveraging a business opportunity is just too slow

Waiting for a funding cycle, big-design-upfront, and a long development cycle results in delayed technical and customer feedback. And more than likely, the learning at the end of this cycle isn't positive. Simply put, it is difficult, if not impossible, to understand and adapt to customer needs and quickly deliver a solution with a traditional development approach.

Introducing the Business Agility Value Stream

Instead, what is needed is a rapid cycle of sensing and responding that helps the Enterprise navigate the unknowns and arrive at a desirable solution **before** the window of opportunity closes. This is the business agility value stream (BAVS), illustrated in Figure 2. And it is explicitly designed to foster rapid learning and enable more favorable business outcomes. By implementing SAFe, enterprises inherently develop the Lean, Agile, and DevOps capabilities that will allow incremental delivery at scale. Although these capabilities are essential, establishing true business agility requires flow to be cultivated and accelerated through the entire BAVS, from sensing an emerging opportunity to deliver the right solution (Figure 8).

Figure 8. A faster way to leverage a new business opportunity

Figure 8. A faster way to leverage a new business opportunity

Steps in the Business Agility Value Stream

While the specific implementation of the BAVS depends on the business context (organizational, market, customer, technology, and solution), the steps and basic

structure are essentially the same. In addition, the knowledge, skills, and behaviors required to succeed with each step of the BAVS are described in the related SAFe Core Competency, as we will see below.

This agility requires all functions, processes, activities, teams, and events from end to end to be aligned and optimized for maximum speed and quality. The following steps form this value stream:

Sense Opportunity

The first step, of course, is to be able to sense the opportunity!

This step requires the SAFe [Organizational Agility](#) core competency, which fosters activities such as:

- Market research
- Analysis of quantitative and qualitative data
- Direct and indirect customer feedback

Direct observation of the customers in the marketplace

Most directly savvy, Lean-thinking leaders Ÿgo see and spend significant time where customers perform work. They return with current, relevant, direct, and specific information about the realities of their products and services and identify opportunities for innovative new solutions.

Fund MVP

An enterprise must be able to respond quickly to these opportunities with nimble funding. [Lean Portfolio Management](#) (LPM) is the core competency needed to support this step.

With LPM, Lean Budgeting provides the ability to quickly allocate sufficient funds to build a Minimum Viable Product (MVP) —an early version of the solution used to evaluate the primary business hypothesis. The “Minimum” in MVP refers to the low cost of the experiment needed to test the hypothesis and

establish solution viability.

These funding decisions become visible and addressed as the new initiative flows through the SAFe Portfolio Kanban system.

Organize Around Value

The next step is to organize—nor reorganize as necessary—to address the new opportunity. An MVP can often be built by existing Agile Teams or Agile Release Trains (ARTs) as the new work find its way into their respective backlogs. However, creating an MVP may also involve modifying existing teams and ARTs. An entirely new development value stream may need to be formed in a more extreme case.

Two core competencies—[Team and Technical Agility](#) and [Organizational Agility](#)—enable this flexibility.

Connect to Customer

Agile development is inherently focused on assuring a direct connection to the customer, and Customer Centricity is the mindset that underpins it. This way of doing business focuses on creating positive experiences for the customer across the enterprise’s complete set of products and services and throughout the entire customer journey.

Design Thinking provides the tools that help teams and ARTs achieve these ideals by empathizing with the user to design the right solution. [Agile Product Delivery](#) is the core competency that enables this connection.

Deliver MVP

The proof is in the doing. Agile teams

and ARTs deliver the MVP iteratively and incrementally, following Lean-Agile practices.

However, there are differences in how teams and trains work on an MVP compared to evolving functionality in a mature solution. There is more risk and uncertainty for a start. Unknowns may manifest in critical areas, including technology choices, implementation strategy, organizational expertise, deployment, operations, customer acceptance, and business benefits. More experimentation and even faster feedback are required. But that is exactly what SAFe is optimized for.

Depending on the scope of the solution, Agile Product Delivery and [Enterprise Solution Delivery](#) are the two core competencies that enable this step.

Pivot or Persevere

The result of the MVP is a set of facts that support a decision regarding whether to proceed with further solution development. If the hypothesis is disproven, the organization accepts the sunk cost and moves on to other business opportunities. If the hypothesis proves beneficial, additional funding follows to enable further development. However, the MVP outcome is not always a simple yes or no. The experiment may yield vital insights that reveal new alternative solutions.

This decision point is a crucial investment milestone and is a critical stage in the portfolio kanban system. Again, Lean Portfolio Management is the core competency that enables this step.

Deliver Value Continuously

A successful MVP that confirms the hypothesis opens the gates to deliver value continuously with additional solution features. This process relies on Agile Product Delivery that fosters iterative and incremental development powered directly by the ART.

Building on DevOps, these practices include optimizing a Continuous Delivery Pipeline that ensures a steady

flow of value and the ability to release on-demand to meet the needs of customers and businesses.

For some organizations, these solutions represent large, significant, and complex applications and cyber-physical systems that require thousands of developers and many capable suppliers to coordinate their efforts within a Solution Train. Enterprise Solution Delivery is the core competency enabling this step in this case.

Learn and Adapt

Learn and Adapt is not the final step. A single initiative rarely determines business outcomes. Instead, the enterprise learns from the BAVS and the process and adapts based on these learnings.

Measurement is an integral part of improvement. As Figure 8 illustrates, three measurement domains—Competency, Flow, and Outcomes—

provide critical perspectives and measures of organizational performance that help identify impediments and opportunities for improvement.

The **Continuous Learning Culture** core competency is the primary driving force behind positive change. A learning organization has a sense of urgency, constantly looking for new business opportunities and improvements in existing processes and solutions. Indeed, the BAVS is an example of continuous learning, enabling continuous innovation. In addition, other forms of learning and adaptation happen through regular Inspect & Adapt events at every level of the SAFe operating system.

Lean-Agile Leadership Enables the BAVS

None of this happens without [Lean-Agile Leadership](#), the foundation of SAFe and the BAVS. The BAVS represents a new way of working for most enterprises, which is substantially different from the status quo and affects most aspects of a modern enterprise.

Lean-Agile leaders view the organization as a dynamic set of business agility value streams that pursue and leverage critical business opportunities. And importantly, they lead the change to arrive at this new state. After that, they focus the organization on successful BAVS execution and improving BAVS performance over time. Only then can business agility be achieved.

Measuring the Business Agility Value Stream

As Figure 9 illustrates, SAFe provides three measurement domains *flow*, *outcomes*, and *competency* suitable for measuring and improving any value stream, including the all-important BAVS.

Figure 8. Three SAFe measurement domains support the goal of business agility

Figure 9. Three SAFe measurement domains support the goal of business agility

Flow metrics help determine how fast the value stream creates and delivers value, represented by flow *distribution, velocity, time, load, efficiency, and predictability*.

Outcomes metrics help ensure the solution delivered benefits the customer and the business. [Value stream KPIs](#) are primarily used to measure these outcomes.

Competency measures evaluate organizational proficiency on two levels:

1. The *SAFe Business Agility Assessment* offers business and portfolio stakeholders a way to assess their overall progress.
2. The individual *SAFe Core Competency Assessments* help teams and ARTs improve their technical and business practices to achieve the portfolio's goal.

See the [Measure and Grow](#) article for more details.

The road to business agility is long and never-ending. Measuring it helps enterprises understand where they are on their journey and reminds them to celebrate small successes.

Summary

Welcome to the age of software and digital, where business agility will be the most significant factor in deciding the winners and losers in the new economy:

Lean-Agile *commercial* businesses will create higher profits, increase employee engagement, and more thoroughly satisfy customer needs

Lean-Agile *nonprofits* will build resilience, sustainability, and the alignment needed to fulfill their mission

Lean-Agile *governments* will deliver systems that better ensure the public's safety, economy, and general welfare

These three market segments depend on delivering innovative business solutions faster and more efficiently than ever. Each will need a dual operating system: a hierarchy for efficiency and scale and a second, customer-centric network operating system that delivers innovative solutions. SAFe's seven core competencies will enable this all-important dual operating system. Those who master these competencies will survive and thrive in the digital age.

Learn More

[1] Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. IT Revolution Press, 2018.

[2] Perez, Carlota. *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages*. Edward Elgar Publishing, 2002.

[3] Kotter, John. *XLR8 (Accelerate): Building Strategic Agility for a Faster-Moving World*. Harvard Business Review Press, 2014.

[4] [https://www.investopedia.com/articles/investing/072115/companies-went-bankrupt-innovation lag](https://www.investopedia.com/articles/investing/072115/companies-went-bankrupt-innovation-lag)

Last update: 13 February 2023

images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English

Scaled Agile Framework



Luck is what happens when preparation meets opportunity.

ÑSeneca

Enablers

Enablers are backlog items that extend the architectural runway of the solution under development or improve the performance of the development value stream.

[Adjust Cookie Settings](#)

Enablers are captured in backlogs as a type of [Epic](#), [Capability](#), [Feature](#), or [Story](#). They

are used primarily for exploration, architecture implementation, refactoring, building infrastructure, and addressing compliance. While their type is unique, they are managed similarly to customer-facing backlog items.

Details

Enablers bring visibility to all the work necessary to support the efficient development and delivery of future business requirements. Enablers are used to explore ideas, improve architecture, strengthen infrastructure, and manage compliance. Since enablers result in the production of tangible outputs, they must be visible. They are treated like all other backlog items—subject to visibility, prioritization, incremental delivery, measurement, and feedback.

Types of Enablers

Enablers can be used to define any activity that improves the value stream in support of foreseeable business needs. These activities generally fall into one of four categories:

Exploration ▷ support research, prototyping, and other activities needed to develop an understanding of customer needs, including the exploration of prospective [Solutions](#) and evaluation of alternatives

Architectural ▷ used to build [Architectural Runway](#), which allows smoother and faster development through the [Continuous Delivery Pipeline](#) (CDP)

Infrastructure ▷ support the creation and optimization of the development and runtime environments that host the systems used to build, validate, deploy, and operate solutions

Compliance ▷ facilitate managing specific compliance activities, including Verification and Validation (V&V), audits and approvals, and policy automation

Creating and Managing Enablers

Enablers exist throughout SAFe and are written and prioritized according to the same rules as their corresponding epics, features, capabilities, and stories.

Enabler Epics ▷ These are written using the “epic hypothesis statement” format, in the same way as business epics. Enabler epics can span multiple [Agile Release Trains](#) (ARTs) and [PIs](#) and are managed via the [Portfolio Backlog](#) and associated Kanban system.

Enabler Features and Capabilities ▷ These are defined by ARTs and [Solution Trains](#) and include a short phrase, benefit hypothesis, and acceptance criteria. They

must be sized to fit within a single PI.

Enabler Stories ☰ Must fit within [Iterations](#) like any story. Although they may not require the user voice format, their acceptance criteria clarify the requirements and support testing.

Architects often define and guide enabler epics, features, and capabilities. They might be [Enterprise Architects](#) supporting the portfolio, [System Architects](#) supporting ARTs, or [Solution Architects](#) supporting [Solution Trains](#). Architects steer enablers through the appropriate Kanban system and backlog, guiding implementation from concept to delivery. [Agile Teams](#) also use enablers; enabler stories emerge locally from their needs and are carried in the [Team Backlog](#).

The following examples illustrate how Agile teams and architects create and manage each of the four enabler types.

Enabling Exploration

Exploration enablers provide work items teams can use to discover requirements and design details. The nature of [Solution Intent](#) is that many requirements begin as variable intent. At the beginning of development, little is known about what the customer needs or how to implement it. Customers themselves often don't understand precisely what they need. Through [Continuous Exploration](#), teams progressively learn which aspects of solution intent should move from variable to fixed.

In an even broader view, there are typically many technical possibilities for implementing an identified business need or opportunity. Those alternatives must be analyzed and are often evaluated through modeling, prototyping, Set-Based Design, or the Lean Startup Cycle. Exploration enablers formalize these activities, assure that the work is visible, and help ensure solution development is closely aligned with the needs of customers and stakeholders.

Enabling Architecture

In SAFe, [Agile Architecture](#) practices produce architectural runway, the underlying technology that enables Agile teams and ARTs to deliver business solutions quickly. But the runway is constantly consumed by business epics, features, capabilities, and stories, so it must be extended for new functionality. Architectural enablers are used to build, extend, and maintain the runway.

Architectural enablers can also address problems with the resiliency of deployed solutions. After implementation, these enablers often reflect [Nonfunctional](#)

[Requirements](#) (NFRs) imposed on future backlog items. NFRs often originate as architectural enablers and grow as a set over time (Figure 1).

Figure 1. Many NFRs appear over time as a result of enablers

Figure 1. Many NFRs appear over time as a result of enablers

Enabling Infrastructure

Agile development requires frequent integration. Agile teams integrate their work and showcase the working solution increment at the [System Demo](#). Similarly, ARTs that are part of a Solution Train integrate their work as frequently as possible during the PI in preparation for [Solution Demos](#). Infrastructure enablers provide the [Continuous Integration](#) and [Continuous Deployment](#) technology that supports this aggressive integration cadence.

The [System Team](#) is integral in defining and building infrastructure enablers that enhance the development environment and streamline the CDP. [Shared Services](#), Operations teams, and Site Reliability Engineering (SRE) leverage infrastructure enablers to deliver [Cloud](#) services that accelerate solution development and solution scalability.

Enabling Compliance

By incrementally building the necessary artifacts in the solution intent over a series of PIs, SAFe supports continuous verification and validation. Verification activities are conducted as part of the development workflow and are often enforced in the Definition of Done (DoD). While the artifacts will satisfy the objective evidence needed at the end of development, they are created iteratively throughout the life cycle. Validation occurs when Product Owners, customers, and end-users participate in ART planning and system demos, validating fitness for purpose.

Enablers are used to support these activities. For example, consider a regulation that requires design reviews and that all actions stemming from those reviews must be documented when completed. A "design review enabler" backlog item would offer evidence of the review, and its DoD would ensure that actions are recorded and resolved according to the Lean Quality Management System (QMS). If needed, the activities themselves could be tracked as enabler stories.

Implementing Enablers Incrementally

Enablers supply critical, foundational technology and insight into the value stream. Consequently, they deserve focused attention in the portfolio, from budgeting and capacity allocation through delivery and ongoing improvement. But because the actual value of enablers is tied to the realization of future business objectives, care must be taken to implement enablers quickly and iteratively. Otherwise, delivering value to customers can be significantly delayed, undermining the fundamental purpose of enablers.

Enablers of all types should be implemented incrementally. However, because architectural and infrastructure enablers often influence the delivery and operation of mission-critical solutions, they deserve special mention here.

The size and demands of architectural and infrastructure enabler work can be overwhelming. So, it's important to remember that they need to be split into features and stories that can be delivered incrementally. This can be difficult, however, as architectural and infrastructure changes can potentially stop the existing system from working until the changes are in place.

The work must be sequenced to ensure the system can continue operating in the current environment while enablers are implemented. That way, teams can continue to work, integrate, demo, and even release new functionality.

There are three ways to approach this [1]:

Case A The enabler is big, but there is an incremental approach to

implementation. The system always runs (operates).

Case B ☰ The enabler is big but cannot be implemented incrementally. The system will need to take an occasional break.

Case C ☰ The enabler is *really* big, and it cannot be implemented incrementally. The system runs when needed.

Examples of incremental patterns are also described in [2], where the legacy subsystems are gradually “strangled” over time, using proven patterns such as asset capture or event interception.

By creating the technology platforms that deliver business functionality, enablers drive better economics. But innovative product development cannot occur without risk-taking. Therefore, initial technology-related decisions cannot always be correct, which is why the Lean enterprise must be prepared to change course occasionally. In these cases, the principle of sunk costs [3] provides essential guidance: Do not consider money already spent. Implementing incrementally allows corrective action before the investment grows too large.

Implementing Enablers Across ARTs and Value Streams

Enabler epics and capabilities can cut across multiple value streams or ARTs. During the analysis stage of the appropriate Kanban system, it is important to determine whether to implement the enabler across all ARTs simultaneously or incrementally (Figure 2).

Figure 2. Two approaches to implementing cross-cutting enablers

Figure 2. Two approaches to implementing cross-cutting enablers

In scenario A, the enabler is implemented first in ART 1, then by the other ARTs in subsequent PIs. This may lessen the impact of the change across the portfolio but can delay the full benefits of a fully implemented enabler. By contrast, scenario B calls for all ARTs to implement the enabler at the same time. This is preferable if the cost of delaying the entire implementation is unacceptably high.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Fowler, Martin. *Strangler Application*.
<http://martinfowler.com/bliki/StranglerApplication.html>

[3] Reinertsen, Donald. *The Principles of Product Development Flow: Second*

Generation Lean Product Development. Celeritas Publishing, 2009.

Last update: 13 January 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US
5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



While building trust gives teams the ability to reconfigure and Òdo the right thing,Ó it is also necessary to make sure that team members know what the right thing is. Team members must all work toward the same goal, and in volatile, complex environments that goal is changeable.

Adjust Cookie Settings

ÑGeneral Stanley
McChrystal, *Team of
Teams*

Team Backlog

The Team Backlog is a Kanban system that is used to capture and manage the user stories and enablers intended to enhance the solution.

This includes stories originating from Features in the ART backlog as well as those arising from the teamÕs local context.

Details

The team backlog holds all the possible work that a team might do to enhance the solution. For example, it contains User [Stories](#), [Enablers](#), and other work items such as improvement stories for corrective actions from the teamÕs [Retrospectives](#) or the [ARTÕs Inspect and Adapt](#).

While itÕs not conceptually complex, some essential aspects make the team backlog critical for Agile development. For example:

It contains all the work for an [Agile Team](#) to advance the solution and aligns all team members to a common goal.

ItÕs a list of wants, not commitments. Items can be estimated (preferable) or not, but it is just an ordered list, and there is no specific time commitment for completion. In other words, the backlog is time-independent, giving the team general flexibility regarding what gets implemented and when.

All team members can enter stories into the backlog.

It has an ownerÑthe Product Owner (PO)Ñwho helps the team manage the challenge of multiple stakeholders who may have divergent views of whatÕs essential.

The PO, with input from the team and other stakeholders, is primarily responsible for creating and maintaining the team backlog. However, any team member can enter an item into the backlog for consideration. The PO prioritizes the backlog, balancing the needs of stakeholders. There are three primary inputs to the team backlog, as Figure 1 illustrates.

Figure 1. Input sources for a team backlog

Figure 1. Input sources for a team backlog

ART Backlog ☰ The [ART Backlog](#) consists of upcoming features planned to be delivered by a train. During [PI Planning](#), teams split the candidate features into stories and tentatively place them into upcoming [Iterations](#). These new stories are maintained in the team backlog.

Team’s local context ☰ The team’s local concerns (other new functionality, defects, refactors, tech debt, and maintenance) are also in the backlog. Since PI planning is high-level, adjustments will likely occur during the PI. Teams using Scrum will probably make adjustments during [Iteration Planning](#), while teams applying Kanban will likely do the same during backlog replenishment.

Other stakeholders ☰ Agile Teams on the ART are not islands, and their backlogs will contain some stories that support other teams’ dependencies and other commitments, including the ART’s [PI Objectives](#). These stories may include spikes for research required to estimate [Features](#), [Capabilities](#), and even [Epics](#).

Moreover, teams get feedback from previous increments, the [System Demo](#), and other groups that may affect the backlog.

Nonfunctional Requirements (NFRs) are persistent qualities that may affect the solution's design, performance, or quality. Since they serve as constraints (or restrictions) for all the team's items, the Big Picture illustrates them at the bottom of the backlog in Figure 1. Due to their importance, teams often automate acceptance tests for NFRs and include them in their definition of done (DoD).

Building and Refining the Backlog

Agile Teams take a continuous, flow-based approach to maintain backlog readiness, so it always contains some stories ready for implementation without significant risk or surprise. Like a neglected garden that grows wild when left unattended for too long, the team backlog becomes unmanageable if not given care and attention. Refining the team backlog includes the following activities:

- Refining stories and establishing acceptance criteria
- The PO regularly prioritizes the team backlog in collaboration with the team and stakeholders
- New stories, including enablers, are discovered and described, and existing ones are changed or removed
- High-priority items are readied by defining acceptance criteria and sizing them to fit within small timeboxes
- Stories that have been around too long, or perhaps are no longer relevant, are removed

Although the PO manages the team backlog, refinement is a collaborative process. It creates a dialogue between the team, *Customers*, and other stakeholders. This refinement breaks down barriers between the business and the development team, eliminating waste, handoffs, and delays. Developing story acceptance criteria increases the clarity of the requirements, leverages the team's collective knowledge and creativity, and creates buy-in and joint ownership.

There is no prescriptive meeting pattern for refining the backlog. Some teams like to do a bit of backlog refinement after their Team Sync. Others prefer weekly refinement sessions or requirements specification workshops, applying *Behavior-Driven Development* (BDD) techniques to help clarify stories. Since multiple teams often collaborate on feature development, new issues, dependencies, and stories will likely arise. Backlog refinement also helps surface problems with the current plan, which may require discussion at the team, PO, or coach syncs.

Managing the Backlog with Kanban

In SAFe, Agile Teams manage their backlog using a Kanban system. The backlog

Kanban system facilitates alignment, visibility, and dependency management. Figure 2 illustrates an example of one team's initial Kanban system.

Figure 2. One Agile Team's initial Kanban board

Figure 2. One Agile Team's initial Kanban board

This Kanban visualizes all active and pending work, workflow states, and work-in-process (WIP) limits. The system is WIP limited; a work item can be pulled into the next step only when the number of items is lower than the WIP limit. A few activities in the Kanban (typically beginning and end) may not be WIP-limited. The team defines and adjusts WIP limits, allowing it to adapt quickly to the flow of complex system development variations.

See the [Applying Kanban in SAFe](#) and [SAFe Team Kanban](#) articles for more information on establishing the team Kanban system.

Balancing Value Delivery and System Health

with Capacity Allocation

Like the ART, every [Agile Team](#) faces the problem of balancing internal work—maintenance, refactors, and technical debt—with the new user stories that deliver more immediate business value. While focusing solely on business functionality may work for a while, this approach will be short-lived as technical debt increases, ultimately slowing development velocity. Avoiding this risk requires continuous investment in evolving the solution’s [Architectural Runway](#) while making customers happy with enhancements, new functionality, and bug fixes. Getting this balance right extends the system’s life, deferring technical obsolescence.

But prioritizing different types of work can be challenging as the PO tries to compare the value of unlike things: defects, refactors, redesigns, technology upgrades, and new user stories. And there is no upper limit to the demand for any of these things.

In collaboration with the team, the PO applies *capacity allocation* (Figure 3) for each item type. Then the PO, team, and [System Architect](#) select the highest-priority backlog items for each capacity allocation slice during planning. Since many stories originate from features, PI planning commitments may predetermine some priorities. However, the PO can prioritize work from the team’s local context by comparing value, size, and logical sequencing. Also, the PO can adjust the allocation percentage for each work item type to address long-term system health and value delivery. Teams should adapt the capacity allocation categories as needed. However, these categories should be consistent across teams in the ART.

Figure 3. Typical examples of capacity allocation categories (user stories, enablers, and maintenance in this case)

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300

Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Imagine a world where product owners, Development, QA, IT Operations, and Infosec work together, not only to help each other, but also to ensure that the overall organization succeeds. By working toward a common goal, they enable the fast flow of planned work into production, while

Adjust Cookie Settings

achieving world-class stability, reliability, availability, and security. [1]

~The DevOps Handbook

DevOps

Note: This is the [home page](#) for the three-part **SAFe DevOps** series. This article is the **first** in the series which introduces the foundational DevOps concepts. The following links provide access to the next articles in the series: [A CALMR Approach to DevOps](#) and [SAFe's DevOps Practice Domains](#).

DevOps is a mindset, culture, and set of technical practices that supports the integration, automation, and collaboration needed to effectively develop and operate a solution.

DevOps is part of the [Agile Product Delivery](#) competency and is a combination of two words: *development* and *operations*. Without DevOps, there is often significant tension between those who build [Solutions](#) and those who support and maintain them. DevOps helps break down organizational silos and develop a [Continuous Delivery Pipeline \(CDP\)](#)—a high-performance innovation engine capable of delivering market-leading solutions at the speed of business.

Figure 1. The Continuous Delivery Pipeline

Figure 1. The Continuous Delivery Pipeline

The goal of DevOps is simple, to deliver value whenever there is a business need. Indeed, teams that successfully adopt DevOps, on average: deploy 208 times more frequently, 106 times faster, experience seven times fewer failures, and recover from incidents 2,604 times faster than low-performing teams. [2]

DevSecOps

DevSecOps is a term that emphasizes the importance of proper information **security** practices in the pursuit of continuous delivery. Because the origins of DevOps did not explicitly include security as a top-level concern (as it did for development and operations), DevSecOps has emerged as a popular term that avoids any risk of security being an afterthought.

The security community has evolved DevOps thinking beyond its development and operations roots. The **State of DevOps Report**—the world’s longest-running and most widely cited DevOps research project—has revealed that an organization’s

security improves when it's wholly integrated into the [Value Stream](#). [3] In one of the most-read DevSecOps articles on the Internet, RedHat reminds us that outdated security practices can undo even the most efficient DevOps initiatives. [4]

The top ten list of software vulnerabilities from the [Open Web Application Security Project \(OWASP\)](#) foundation has become one of the most relied-on tools for fostering collaboration between development, operations, and security teams. [5]

The US Air Force pioneered the DevSecOps Platform (DSOP) initiative, demonstrating that combining advanced DevOps and security practices can provide some of the most highly regulated organizations in the world with "plug and play" software factories and radically streamlined delivery processes.

Thanks to these contributions, security has become deeply ingrained in DevOps culture. As a result, *DevOps* and *DevSecOps* have come to mean the same concept for all practical purposes. Each implies a set of blended practices from multiple domains—development, operations, security, infrastructure, architecture, and so on throughout the value stream—that work together to enable collaboration, speed, quality, and safety.

Figure 2. Security is built into DevOps in SAFe

SAFe carries this sentiment forward, treating security as a primary concern. In SAFe, to say "DevOps" means "DevSecOps." Protecting customers, employees, citizens, soldiers, families, and businesses is not something we choose to do or not do in DevOps. It is simply in our DNA. As such, modern security practices shine through in many areas of SAFe, including the Big Picture, Framework guidance, courseware, assessments, [Extended SAFe Guidance](#) articles, and more.

These are just a few examples of how the DevSecOps movement has lifted DevOps to new standards of excellence.

Agile Release Trains (ARTs) are the primary value delivery construct in SAFe. Each ART has all the skills necessary to build and release the solution, including those

responsible for **Security**, Compliance, Quality Assurance (QA), Testing, and Verification and Validation (V&V). Each increment the ARTs builds assesses the viability of the current solution and its progress toward security, quality, and compliance, providing early feedback on the system's ultimate fitness for use. Second, specifications are created early and evolve in small batches, with faster feedback on decisions and the opportunity for continuous review and assessment. ARTs cannot implement security through *inspection; it must be built into the solution* during each iteration. Security testing should shift left to prevent vulnerabilities and be automated to increase the speed and accuracy of compliance.

Details

DevOps makes continuous delivery possible. Indeed, enterprises wishing to deliver value to customers and stakeholders continually should master the DevOps **mindset** and technical practices. These skills are critical in this era of constant digital disruption and innovation. Achieving continuous delivery, however, at scale, is not easy. SAFe's approach to DevOps helps enterprises navigate these complexities.

A Paradigm Shift

IT organizations worldwide are troubled by a core, chronic conflict: technology delivery processes rely on teams with *seemingly opposing* goals and incentives. [1] **Agile Teams** deliver changes quickly to keep pace with business needs. Operations regulate the flow of changes to maintain the stability of solutions that run the business. Security teams institute policies to prevent changes from introducing vulnerabilities that can cause data breaches.

Keeping pace requires a new delivery system — a "software factory" — which aligns teams and increases delivery speed while simultaneously increasing solution quality, security, and stability. Only then can the needs of customers and teams be predictably and effectively met.

Note: "Software factory" is an increasingly popular term for this new delivery system. In his SAFe [Community Contributions](#) article, Peter Vollmer, Distinguished Technologist at Micro Focus, describes a software factory as "standardized tooling and engineering services" that supports and enhances [the value stream]. [6]

These software factories are integrated sets of tooling, services, data, and processes that help move products through the plan, build, test, and release cycles. [7] The US Department of Defense (DoD) maintains a growing ecosystem of software factories, leveraging a common DevSecOps Platform (DSOP) to rapidly deliver specialized digital products and services. [8] Regardless of the term used to describe the system, enterprises leverage DevOps to achieve this level of sophistication in their value streams.

Unfortunately, most IT organizations do not natively support this kind of system. Their processes and policies are optimized to prevent frequent changes to production systems, not enable them. Therefore, a paradigm shift is needed. Just as Agile represents a paradigm shift in how *we work*, DevOps represents a similar shift in how *we build*. Leveraging DevOps to usher in a new way of building digitally-enabled solutions is the key to transforming outdated development life cycles into CDPs.

Continuous Learning and Experimentation

CDPs are the result of applying DevOps effectively to value streams. And value streams need to behave differently than they did in the old model because today's technology delivery objectives differ.

Enterprises must release features faster than ever to remain relevant in their markets. But out-deploying the competition is not the goal. Out-learning them is. And that learning comes from understanding the new functionality's value in the market. Since features have no value until released, enterprises must constantly build, measure, and learn to evolve digital solutions that quickly attract and retain customers. Figure 3 shows that SAFe's CDP operates as a closed-loop system that fosters rapid, low-risk experimentation and continuous learning about customers' needs, habits, and preferences.

Figure 3. The CDP as a Continuous Learning Loop

Figure 3. The CDP as a Continuous Learning Loop

This relentless learning and experimentation engine is starkly different than traditional delivery processes. Enabling it requires a different mindset, skills, and tools across the entire value stream. Large batches, siloed teams, handoffs, monolithic architectures, change review boards, politics, and heroics have no place here. Instead, this new system needs to be guided by shared values, cross-functional collaboration, objective measurements, automation, and modern technical practices.

Enter DevOps.

Figure 4 illustrates how DevOps enables the CDP. It does this by supplying the *mindset*, *practices*, and *tooling* required to foster rapid delivery and learning at every step.

Figure 4. DevOps enables the CDP

Figure 4. DevOps enables the CDP

At its core, DevOps is a mindset that guides behavior and decision-making throughout the value stream. SAFe's CALMR approach to DevOps embodies this mindset, is central to the figure above, and permeates all aspects of the CDP. DevOps technical skills, practices, and tooling evolve and sustain solutions directly. In SAFe, *practice domains* represent this knowledge within the inner rings of the CDP model shown in Figure 4.

Measuring and Managing DevOps Maturity

Measuring DevOps performance and tracking incremental progress are essential to building a thriving DevOps culture.

The SAFe DevOps Health Radar (Figure 5) is a tool that helps ARTs and Solution Trains optimize their value stream performance. It provides a holistic DevOps health check by assessing the maturity of the CDP's four aspects and 16 activities.

The Health Radar measures baseline maturity at any point in a DevOps transformation and help guide fast incremental progress.

DevOps Health Radar

Figure 5. The SAFe DevOps Health Radar

Figure 5. The SAFe DevOps Health Radar

Download the free DevOps Health Radar assessment [here](#). Agility Health, a Scaled Agile partner, also offers an [online version of this assessment](#).

More in the DevOps Series

Article 1: DevOps Home Page (this page)

Article 2: [A CALMR Approach to DevOps](#)

Article 3: [SAFe's DevOps Practice Domains](#)

Learn More

[1] Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

[2] Accelerate © State of DevOps 2019. <https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>

[3] 2019 State of DevOps Report. <https://puppet.com/resources/report/2019-state-of-devops-report>

[4] What is DevSecOps? <https://www.redhat.com/en/topics/devops/what-is-devsecops>

[5] OWASP Top 10 Application Security Risks. <https://owasp.org/www-project-top-ten/>

[6] Accelerating Flow with DevSecOps and the Software Factory.
<https://www.scaledagileframework.com/accelerating-flow-with-devsecops-and-the-software-factory/>

[7] Why a Software Factory Is Key to Your DevOps Success.
<https://techbeacon.com/devops/why-software-factory-key-your-enterprise-devops-success>

[8] Software Factories. <https://software.af.mil/software-factories/>

Last update: 14 March 2023

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

[Download SAFe](#)
[Posters & Graphics](#)
[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)
[About Certification](#)
[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)
[Partner Directory](#)

GET SOCIAL

[Twitter](#)
[Linkedin](#)
[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)
[Permissions Form](#)
[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Simplicity—the art of maximizing the amount of work not done—is essential.

—The Agile Manifesto

Essential SAFe

Essential SAFe provides the minimal elements necessary for Agile Release Trains to deliver solutions and is the simplest starting point for implementation.

As a common SAFe starting point, Essential SAFe excludes Enterprise Solution Delivery and Portfolio concerns.

The [Agile Release Train \(ART\)](#) is the heart of Essential SAFe. ARTs are virtual organizations formed to span functional boundaries, eliminate unnecessary handoffs and steps, and accelerate value delivery by adopting SAFe's [Lean-Agile Principles](#) and practices.

The Essential SAFe configuration (Figure 1) includes the following constructs:

The Foundation element, which includes the [Lean-Agile Mindset](#), [Core Values](#), [SAFe Principles](#), [Implementation Roadmap](#), the role of the [SPC](#), and two core competencies:

[Lean-Agile Leadership](#) ☐ Describes how Lean-Agile Leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential.

[Continuous Learning Culture](#) ☐ Describes a set of values and practices that continually encourage individuals and the enterprise as a whole to increase knowledge, competence, performance, and innovation.

Two delivery core competencies:

[Team and Technical Agility](#) ☐ Describes the critical skills and Lean-Agile principles and practices that high-performing Agile teams and Teams of Agile teams use to create high-quality solutions for their customers.

[Agile Product Delivery](#) ☐ A customer-centric approach to defining, building, and releasing a continuous flow of valuable products and services to customers.

The essential level roles, artifacts, and events

A minimal spanning palette

Figure 1. Essential SAFe

Figure 1. Essential SAFe

Essential SAFe provides a starting point for Implementing SAFe, including the ten critical success factors needed for an ART to realize the majority of the Framework's benefits (see the last section of this article.)

Details

Essential SAFe provides the relevant roles, artifacts, events, and mindset for ARTs to deliver one or more desirable, feasible, viable, and sustainable Solutions. The ART's long-lived, flow-based, self-organizing nature powers SAFe and ultimately enables [Business Agility](#). ARTs are typically virtual, spanning organizational and geographic boundaries; others follow a line of business or product line management reporting structure.

Highlights

The highlights of Essential SAFe include the following:

Agile Release Train (ART) Δ a long-lived team of Agile teams that incrementally develops, delivers and often operates one or more solutions in a value stream.

Continuous Delivery Pipeline Δ Describes the workflows, activities, and automation needed to release value to the end user constantly.

Customer Centricity Δ A mindset and a way of doing business that focuses on creating positive customer experiences.

Design Thinking Δ An iterative solution development process that assures solutions are desired by customers and users while ensuring the solution is feasible, economically viable, and sustainable throughout its lifecycle.

Lean UX Δ Lean User Experience is a mindset, culture, and process that embraces Lean-Agile methods. It implements functionality in minimum viable increments and determines success by measuring results against a benefit hypothesis.

PI (Planning Interval) Δ is a fixed timebox during which an Agile Release (ART) Train delivers continuous value to achieve its PI Objectives.

Iterations Δ Fixed-length timeboxes that provide the development cadence for Agile teams building solutions.

Innovation and Planning (IP) Iteration Δ Provides the teams with an opportunity for exploration and innovation, dedicated time for planning, and learning through informal and formal channels.

SAFe Scrum Δ A lightweight process for Agile Teams to deliver value continuously. Most SAFe teams use SAFe Scrum to facilitate their work, including Built-in quality practices.

SAFe Team Kanban Δ a Lean-Agile method that helps Agile Teams facilitate the flow of value by visualizing workflow, establishing Work in Process (WIP) limits, delivering value continuously, measuring throughput, and continually improving their process.

Built-In Quality Δ a set of practices established to ensure that the outputs of business functions or technology teams meet appropriate quality standards throughout the process of creating customer value.

DevSecOps Δ a mindset, culture, and set of technical practices. It provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a system.

ART Flow Δ a state where an Agile Release Train (ART) delivers a continuous flow of value to the customer.

Team Flow Δ a state in which Agile Teams deliver a continuous flow of value to the customer.

Roles

ARTs are self-managing and self-organizing teams of Agile teams that plan, commit, and execute together. Agile Team and ART roles help guide and direct the ART, aligning them to a shared mission and providing the necessary Lean governance.

ART Roles

Product Management Δ responsible for defining and supporting the building of desirable, feasible, viable, and sustainable products that meet customer needs over the product-market lifecycle.

System Architect Δ an individual or small cross-discipline team that applies **Principle #2, Apply Systems Thinking**. They define the overall architecture of the system, help identify Nonfunctional Requirements (NFRs), determine the significant elements and subsystems, and help design the interfaces and collaborations among them.

Release Train Engineer (RTE) Δ a servant leader and the chief Scrum Master/Team Coach for the train. The RTE facilitates optimizing the flow of value by ensuring the ART events and artifacts function correctly, including the ART Kanban, Inspect & Adapt (I&A) workshop, ART Sync, and PI Planning.

Business Owners Δ a small group of stakeholders with the business and technical responsibility for fitness for use, governance, and return on investment (ROI) for a **Solution** developed by an ART. They are primary stakeholders in the ART and actively participate in ART events.

Team Roles

Agile Teams Δ a cross-functional group of ten or fewer individuals who can define, build, test, and deploy an increment of value in a short time box. Each ART comprises 5 Δ 15 Agile teams and includes the roles and infrastructure necessary to deliver fully working and tested business solutions.

Product Owner (PO) Δ the content authority for the team backlog is responsible for defining stories and prioritizing the backlog.

Scrum Master/Team Coach (SM/TC) Δ a servant leader and Agile team coach who helps the team remove impediments, facilitates team events, and fosters an environment for high-performing teams.

Events

Essential SAFe has multiple activities to help coordinate the ARTs and teams:

ART Events

PI Planning Δ a cadence-based, face-to-face planning event that serves as the heartbeat of the ART, aligning all the teams on the ART to the shared mission.

System Demo Δ provides an integrated view of new features from the most recent iteration delivered by all the teams in the ART. Each demo includes ART stakeholders with an objective measure of progress during a PI.

Inspect & Adapt Δ a significant event where the current state of the solution is demoed and evaluated. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop.

Coach Sync Δ helps coordinate the dependencies of the ARTs and provides visibility into progress and impediments.

[Product Owner \(PO\) Sync](#) Δ provides visibility into how well the ART is progressing toward meeting the ART PI objectives, discusses problems or opportunities with feature development, and assesses any scope adjustments.

[ART Sync](#) Δ combines the Coach Sync and PO Sync into a single event for an ART.

Team Events

SAFe Scrum teams and some SAFe Kanban teams manage their process with a series of regular events. These typically include the following:

[Iteration Planning](#) Δ a team event in which an Agile team determines the iteration goals and how much of the team backlog they can commit to during an upcoming iteration. Team capacity determines the number of stories and enablers that are selected.

[Iteration Review](#) Δ a cadence-based event at the end of each iteration in which the team reviews the previous increment's results and adjusts the team backlog based on feedback.

[Iteration Retrospective](#) Δ an event held at the end of the iteration for the Agile team to review its practices and identify ways to improve. The retrospective applies qualitative and quantitative information presented during the iteration review.

[Backlog refinement](#) Δ an event is held once or twice during the iteration to refine, review, and estimate future stories and enablers in the team backlog.

[Team Sync](#) Δ a short meeting (usually 15 minutes or less), typically held about daily, to inspect progress toward the iteration goal, communicate, and adjust upcoming planned work.

Artifacts

The following Essential SAFe items help the ART coordinate and deliver work:

ART Artifacts

[Features](#) Δ services that fulfill stakeholders' needs, sized to fit within the PI. Each includes a name, benefits hypothesis, and acceptance criteria.

[Enabler Features](#) Δ supports the activities needed to extend the [Architectural Runway](#) to provide future business functionality, including exploration, architecture, infrastructure, and compliance.

[ART Epics](#) Δ epics that a single ART can deliver.

[ART PI Objectives](#) Δ describe the specific business and technical goals the ART intends to achieve in the upcoming PI.

[ART Backlog](#) Δ a holding area for upcoming Features intended to address user needs and deliver business benefits for a single Agile Release Train (ART). It also contains the enabler features necessary to build the [Architectural Runway](#).

[Vision](#) Δ describes the future state of the solutions under development. It reflects the customer and stakeholder needs and the features proposed to meet them.

Architectural Runway Ø consists of the existing code, components, and technical infrastructure necessary to implement prioritized, near-term features without excessive redesign and delay.

Solution Ø a product, service, or system ARTs deliver to the enterpriseÕs internal or external customers.

Solution Context Ø describes how the system will interface and be packaged and deployed in its operating environment.

Team Artifacts

Stories Ø provides short descriptions of a small piece of desired functionality written in the userÕs language.

Enabler stories provide the exploration, infrastructure, architecture, or compliance groundwork that another story or feature needs.

Team PI Objectives Ø summarized description of the specific business and technical goals an Agile team intends to achieve in the upcoming PI.

Team Backlog Ø consists of user and enabler stories; most are identified during PI planning and backlog refinement events.

Spanning Palette

Vision Ø describes a future view of the solution to be developed, reflecting customer and stakeholder needs and the Features and Capabilities proposed to address those needs.

Roadmap Ø communicates planned ART and value stream deliverables and milestones over a timeline.

System Team Ø a unique Agile team that assists in building and using the continuous delivery pipeline and, where necessary, validating full end-to-end system performance.

Ten Critical Success Factors

SAFe has proven to scale in all situations, from complex software and systems development to bond trading and medical devices to memory chips and fighter aircraft. But, with such a robust Framework, the question becomes: how closely does an organization need to follow various SAFe practices to get the desired result?

Also, when diagnosing SAFe implementation problems, it sometimes becomes apparent that enterprises may have skipped or stopped performing some of these critical practices. To support these challenges, the following Ten Critical Success Factors (Figure 2) highlight the minimal SAFe elements necessary for success.

A large, faint watermark-like diagram titled "Ten Critical Success Factors" is visible in the background. It features a central circle with the number "10" and ten arrows pointing from it to a ring of ten boxes, each containing one of the ten factors.

Figure 2. Ten critical success factors

#1 ⚡ Lean-Agile Principles

SAFe practices are grounded in fundamental [Lean-Agile Principles](#). As organizations adopt SAFe, their continuous improvement activities find even better working methods. These principles guide those improvement efforts and ensure the adjustments move toward the Őshortest sustainable lead time, with the best quality and value to people and society.Ó

#2 ⚡ Real Agile Teams and Trains

Real [Agile Teams](#) and [ARTs](#) are fully cross-functional and can define, build and test their work. They have everything and everyone necessary to produce a working, tested increment of the solution. They are self-organizing and self-managing, enabling value to flow more quickly with minimal overhead.

#3 ⚡ Cadence and Synchronization

Cadence provides a rhythmic pattern, which offers a steady heartbeat for the development process. It makes routine those things that can be routine. Synchronization allows multiple perspectives to be understood and resolved at the same time. For example, synchronization pulls the various assets of a system together to assess solution-level viability.

#4 ⚡ PI Planning

No event is more powerful in SAFe than [PI planning](#) which provides the rhythm for the ART and connects strategy to execution by ensuring business and technology alignment. PI Planning is where *the people who do the work plan the work*. Aligning the entire ART with a common vision and goal creates energy and a shared

sense of purpose.

#5 ⚡ Customer Centricity, DevOps, and Release on Demand

SAFe enterprises create a positive [customer experience](#) across their products and services. They adopt a [DevOps](#) mindset, culture, and applicable technical practices to enable more frequent and higher-quality releases as [the market demands](#). These practices provide faster validation of hypotheses and produce greater profits, increased employee engagement, and more satisfied customers

#6 ⚡ System Demo

The primary measure of the ART's progress is the objective evidence provided by a working solution in the [System Demo](#). Every two weeks, the entire system—the integrated work of all teams on the train for that iteration—is demoed to the train's stakeholders. Stakeholders provide the feedback the train needs to stay on course and take corrective action. This feedback replaces other forms of governance that create additional work and slow flow.

#7 ⚡ Inspect and Adapt

[Inspect and Adapt](#) is a significant event held every PI. It is a regular time to reflect, collect data, and solve problems. The inspect and adapt event assembles teams and stakeholders to assess the solution and define improvements and actions needed to increase the following PI's velocity, quality, and reliability.

#8 ⚡ IP Iteration

The [Innovation and Planning Iteration](#) occurs every PI and serves multiple purposes. It is an estimating buffer for meeting PI objectives and provides dedicated time for innovation, continuing education, PI Planning, and Inspect and Adapt. IP Iteration activities enable many Lean-Agile principles that foster business agility.

#9 ⚡ Architectural Runway

[Architectural Runway](#) consists of the existing code, components, and technical infrastructure necessary to implement high-priority, near-term features without excessive delay and redesign. Insufficient investment in the architectural runway slows the train and makes the ART's delivery less predictable.

#10 ⚡ Lean-Agile Leadership

For SAFe to be effective, the enterprise's leaders and managers must take responsibility for Lean-Agile adoption and success. Executives and managers must become [Lean-Agile leaders](#) who are trained and then become trainers in these leaner ways of thinking and operating. Without leadership taking responsibility for the implementation, the transformation will likely fail to achieve the full benefits.

Learn More

[1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.

Last update: 30 November 2022

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

- [CONTACT US](#)
- 5400 Airport Blvd., Suite 300
- Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm

Weekends: CLOSED

Become a Trainer

[Privacy Policy](#)

[Cookie Policy](#)

[Your California](#)

[Consumer Rights](#)

English

[Scaled Agile Framework](#)

SAFe Advanced Topics

SAFe advanced topic articles are categorized into three areas:

[Extended SAFe Guidance](#)

[Community Contributions](#)

[SAFe Beyond IT](#)

Community Contributions

A
[Adjust Cookie Settings](#)

Accelerating Flow with DevSecOps and the Software Factory

Achieving Business Agility with Business Engagement

Agile People Operations

C

Continuous Delivery

D

Design for Testability: A Vital Aspect of the System Architect Role in SAFe

E

Enterprise Backlog Structure and Management

Enterprise Workflow with the SAFe Portfolio Kanban

Enterprise Technology Strategy Formulation

H

Hardware Teams in SAFe

HR Playbook for a Successful SAFe Implementation

Human-Centered Design with SAFe

I

Implementation Strategies for Business Epics

Improving Strategic Investment Impact by Limiting Portfolio WIP

Invitation-based SAFe implementation

L

Lean UX and the SAFe Program Increment Life Cycle

Lean-Agile Financial Planning Original White Paper

M

Mixing Agile and Waterfall Development

P

Portfolio Planning Tool

R

[Right-Sizing Features for SAFe Program Increments](#)

S

[SAP Delivery Agility](#)

[Six SAFe Practices for ÔS-SizedÕ Teams](#)

[Startup SAFe](#)

T

[Technology Business Management and SAFe](#)

[The Role of PI Objectives](#)

Extended SAFe Guidance

A

[Agile Architecture in SAFe](#)

[Agile Contracts](#)

[Agile Software Engineering Landing Page](#)

[Agile Testing](#)

[Agile Workspaces](#)

[Applying Kanban in SAFe](#)

[Applying SAFe to Hardware Development](#)

B

[Balancing the Dual Operating System](#)

[Behavior-Driven Development](#)

C

[CapEx and OpEx](#)

[Complying with Regulatory and Industry Standards](#)

D

[Distributed PI Planning](#)

[Domain Modeling](#)

E

[Evolving Role of Managers in Lean-Agile Development](#)

F

[Facilitating SAFe Assessments](#)

I

[Innovation Accounting in SAFe](#)

O

[Organizing Agile Teams and ARTs: Team Topologies at Scale](#)

R

[Recommended Reading](#)

[Refactoring](#)

S

[SAFe for Marketing](#)

[SAFe Requirements Model](#)

T

[Test-Driven Development](#)

U

[User Story Primer White Paper](#)

W

[Working Successfully in Agile with Remote Team Members](#)

[What's new in SAFe 6.0?](#)

SAFe Beyond IT

Last update: 11 March 2023

FRAMEWORK

[Download SAFe Posters & Graphics](#)

[Watch and download SAFe videos and presentations](#)

[Blog](#)

TRAINING

[Course Calendar](#)

[About Certification](#)

[Become a Trainer](#)

PARTNER

[Becoming a Partner](#)

[Partner Directory](#)

GET SOCIAL

[Twitter](#)

[Linkedin](#)

[YouTube](#)

CONTENT & TRADEMARKS

[FAQs on how to use SAFe content and trademarks](#)

[Permissions Form](#)

[Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

[Privacy Policy](#)
[Cookie Policy](#)
[Your California Consumer Rights](#)

English

Scaled Agile Framework



Future product development tasks can't be predetermined. Distribute planning and control to those who can understand and react to the end results.

—Michael Kennedy,
Product Development for the Lean Enterprise

There is no magic in SAFe . . . except

Adjust Cookie Settings

*maybe for PI
Planning.*

ÑAuthors

PI Planning

PI Planning is a cadence-based event for the entire ART that aligns teams and stakeholders to a shared mission and vision.

PI planning is essential to SAFe: If you are not doing it, you are not doing SAFe.

Introduction to PI Planning: A Quick Overview

Details

The Agile Manifesto states, “The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.” SAFe takes this to the next level with PI planning.

Where possible, everyone is face-to-face (virtually or physically), and these large-scale PI planning events now occur within many enterprises worldwide. They have clearly shown real financial ROI, not to mention the intangibles that happen when the team of Agile teams creates a social construct that is personally and collectively rewarding.

It may not always be practical for the entire [Agile Release Train \(ART\)](#) to collocate; however, in our current times, COVID-19 has created a situation where this isn’t an option. While physical face-to-face planning has benefits, the unwritten SAFe “rule” is that the people who do the work plan the work. Real-time, concurrent, virtual, face-to-face planning has now proven effective when physical presence is not possible. Indeed many ARTs have been flourishing in creating a hybrid situation where several teams join remotely, as shown below in Figure 1.

Find a Course:

The advanced topic article, [Distributed PI Planning with SAFe](#), provides additional guidance and considerations for successfully managing these scenarios.

Figure 1. Face-to-face PI planning. Remote teams are planning at the same time using video conferencing.

PI Planning has a standard agenda that includes a presentation of business context and [vision](#), followed by team planning breakouts—where the teams create their [Iteration](#) plans and objectives for the upcoming [PI](#). Facilitated by the [Release Train Engineer \(RTE\)](#), this event includes all members of the ART and occurs within the [Innovation and Planning \(IP\) Iteration](#). Holding the event during the IP iteration avoids affecting the scheduling or capacity of other iterations in the PI. PI Planning takes two days, although the ART can extend this timebox to accommodate planning across multiple time zones.

Business Benefits of PI Planning

PI planning delivers many business benefits, including:

- Establishing face-to-face communication among all team members and stakeholders
- Building the social network the ART depends upon
- Aligning development to business goals with the business context, vision, and [Team and ART PI objectives](#)
- Identifying dependencies and fostering cross-team and cross-ART collaboration
- Providing the opportunity for just the right amount of architecture and [Lean User Experience \(UX\)](#) guidance
- Matching demand to capacity and eliminating excess Work in Process (WIP)
- Fast decision-making

Inputs and Outputs of PI Planning

Inputs to PI planning include:

- Business context (see “[Content readiness](#)” below)

Roadmap and vision

Highest priority [Features](#) of the ART Backlog

A successful PI planning event delivers two primary outputs:

Committed PI objectives □ Each team creates a set of [SMART](#) objectives with the business value assigned by the Business Owners.

ART planning board □ Highlighting the new feature delivery dates, feature dependencies among teams, and relevant milestones

Preparation

PI planning is a significant event that requires preparation, coordination, and communication. It is facilitated by the RTE and event attendees, including [Business Owners](#), [Product Management](#), [Agile Teams](#), [System](#) and [Solution Architects](#), the [System Team](#), and other stakeholders. The RTE must schedule all PI planning in advance to be well prepared. The active participation of Business Owners in this event provides an essential [Guardrail](#) on budgetary spending.

For the event to be successful, preparation is required in three major areas:

1. **Organizational readiness**
2. **Content readiness**
3. **Logistics readiness**

The following sections describe these three areas.

Organizational Readiness

Before PI planning, there must be strategy alignment among participants, stakeholders, and Business Owners. Critical roles are assigned. To address this in advance, however, event organizers must consider the following:

Planning scope and context □ Is the planning process's scope (product, system, technology domain) understood? Do we know which teams need to plan together?

Business alignment □ Is there reasonable agreement on priorities among the Business Owners?

Agile teams □ Do we have Agile teams? Are there dedicated team members and an identified [Scrum Master/Team Coach](#) and [Product Owner](#) for each team?

Content Readiness

It's equally important to have a clear vision and context so that the right stakeholders can participate. Therefore, the PI planning must include the following:

Executive briefing ☰ A briefing that defines the current business context

Product vision briefing(s) ☰ Briefings prepared by Product Management, including the top 10 features in the [ART Backlog](#)

Architecture vision briefing ☰ A presentation made by the CTO, [Enterprise Architect](#), or System Architect to communicate new [Enablers](#), features, and [Nonfunctional Requirements](#) (NFRs)

Logistics Readiness

Preparing an event to support a large number of attendees isn't trivial. This prep can include securing and preparing the space for physically collocated planning. For remote attendees or a fully distributed PI Planning, this also includes investment in the necessary technical infrastructure. Considerations include:

Locations ☰ Each location where planning takes place needs preparation in advance.

Technology and tooling ☰ Real-time access to information and tooling to support distributed planning or remote attendees

Communication channels ☰ Primary and secondary audio, video, and presentation channels must be available

Standard Agenda

The event follows an agenda similar to Figure 2. Descriptions of each item follow. For guidance on adapting this agenda to support planning across multiple time zones, refer to the advanced topic article, [Distributed PI Planning with SAFe](#).

Figure 2. Standard two-day PI planning agenda

Figure 2. Standard two-day PI planning agenda

Day 1 Agenda

Business context Ⓛ A Business Owner or senior executive describes the current state of the business, shares the [Portfolio Vision](#), and presents a perspective on how effectively existing solutions address current customer needs.

Product/solution vision Ⓛ Product Management presents the current vision (typically represented by the top ten or so upcoming features). They highlight changes from the previous PI planning event and any relevant milestones.

Architecture vision and development practices Ⓛ The System Architect presents the architecture vision. Also, a senior development manager may introduce Agile-supportive changes to development practices, such as test automation, [DevOps](#), [Continuous Integration](#), and [Continuous Deployment](#), which the teams will adopt in the upcoming PI.

Planning context and lunch Ⓛ The RTE presents the planning process and expected outcomes.

Team breakouts #1 Ⓛ In the breakout, teams estimate their capacity for each [Iteration](#) and identify the backlog items they will likely need to realize the features. Each team creates draft plans, visible to all, iteration by iteration.

During this process, teams identify risks and dependencies and draft their initial team PI objectives. The PI objectives typically include Ôuncommitted objectives,Õ which are goals built into the plan (for example, stories that have been defined and included for these objectives) but are not committed to by the team because of too many unknowns or risks. Uncommitted objectives are *not* extra things to do in case there is time. Instead, they increase the reliability of the plan and give management an early warning of any objectives that the ART may not be able to deliver. The teams also add the features and associated dependencies to the ART Planning Board, as shown in Figure 3.

Figure 3. ART planning board showing features and dependencies

Figure 3. ART planning board showing features and dependencies

Draft plan review During the tightly timeboxed draft plan review, teams present key planning outputs, which include capacity and load, draft PI objectives, potential risks, and dependencies. Business Owners, Product Management, and other teams and stakeholders review and provide input.

Management review and problem-solving Draft plans likely present challenges

like scope, people and resource constraints, and dependencies. During the problem-solving meeting, management may negotiate scope changes and resolve other problems by agreeing to various planning adjustments. The RTE facilitates and keeps the primary stakeholders together for as long as necessary to make the decisions needed to reach achievable objectives.

Solution Trains often hold an additional management review and problem-solving workshop after the first day of planning to address cross-ART issues. Alternatively, the RTEs of the involved trains may talk with each other to discuss the problems for the ART's specific management review and problem-solving meeting. The **Solution Train Engineer (STE)** helps facilitate and resolve issues across the ARTs.

Day 2 Agenda

Planning adjustments D The next day, the event begins with management presenting changes to the planning scope, people, and resources.

Team breakouts #2 D Teams continue planning and making the appropriate adjustments. They finalize their objectives for the PI, to which the Business Owners assign business value, as shown in Figure 4.

Figure 4. A team's PI objectives sheet with assigned business value

Final plan review and lunch D All teams present their plans to the group during this session. At the end of each team's time slot, the team states its risks and impediments and provides the risks to the RTE for use later in the ROAMing exercise.

The team then asks the Business Owners if the plan is acceptable. If the plan is accepted, the team brings their team PI objective sheet to the front of the room so everyone can see the aggregate objectives unfold in real-time. If the Business Owners have concerns, teams can adjust the plan to address the identified issues. The team then presents its revised plan.

ART PI Risks During planning, teams have identified risks and impediments that could impact their ability to meet their objectives. These are resolved in a broader management context before the whole train. One by one, the risks are discussed and addressed with honesty and transparency and then grouped into one of the following categories:

Resolved The teams agree that the risk is no longer a concern

Owned Someone on the train owns the risk since it cannot be addressed during PI planning

Accepted Some items are simply facts or potential problems that must be understood and accepted

Mitigated Teams identify a plan to reduce the impact of the risk

Confidence vote Once ART PI Risks have been addressed, teams vote on their confidence in meeting their team PI objectives

Each team conducts a vote using their fingers (fist of five) or a digital tool for remote events. If the average is three fingers or above, then management should accept the commitment. If it's less than three, the team reworks its plan. Anyone voting two fingers or fewer should be allowed to voice their concerns. These concerns might add to the risk list, require replanning, or provide information. Once each team has voted, it's repeated for the entire ART, with everyone expressing their confidence in the collective plan, as illustrated in Figure 5.

Figure 5. Confidence vote for an ART

Plan rework ☐ If necessary, teams adjust their objectives until they have high confidence. This additional planning is one occasion where alignment and commitment are valued more highly than adhering to a timebox.

Planning retrospective and moving forward ☐ Finally, the RTE leads a brief retrospective for the PI planning event to capture what went well, what didn't, and what to do better next time, as shown in Figure 6.

Figure 6. PI planning retrospective

Next steps Typically, a discussion about the next steps, along with final instructions to the teams, follows, including:

- Cleaning up the rooms used for planning (if applicable)
- Entering the team PI objectives and stories in Agile lifecycle management (ALM) tooling
- Reviewing team and ART events calendars
- Determining [Iteration Planning](#) and Team Sync locations and timing

After the planning event, the RTE and other ART stakeholders summarize the individual team PI objectives into a set of ART PI objectives (Figure 7) and use this to communicate externally and track progress toward the goals.

Product Management uses the ART PI objectives to refine the roadmap, improving the forecast for the following two PIs.

The ART Planning board is often used during the [Coach Sync](#) to track dependencies. It may or may not be maintained (manually) after planning is complete. A digital tool for managing dependencies facilitates their follow-up.

Teams leave the PI planning event with a prepopulated iteration backlog for the upcoming PI. They take their team's PI objectives, iteration plans, and risks to their regular work area. ART risks remain with the RTE, which ensures that the people responsible for owning or mitigating a risk have captured the information and are actively managing the risk.

Most importantly, the ART executes the PI, tracking progress and adjusting as necessary as new knowledge emerges. Execution of the PI begins with all the teams conducting planning for the first iteration, using their PI plans as a starting point. It offers fresh input for the iteration planning processes that follow. Since the iteration plans created during PI Planning did not consider detailed story-level acceptance criteria, the team will likely adjust the first and subsequent iteration plans.

Figure 7. ART PI objectives

Figure 7. ART PI objectives

Solution Train PI Planning

This article focuses on the planning activities of a single ART. However, large [Value Streams](#) may contain multiple ARTs and suppliers. In this case, the Solution Train provides coordination using [Pre-Plan](#) and [Coordinate and Deliver](#) activities.

Learn More

- [1] Knaster, Richard, and Dean Leffingwell. *SAFe 5.0 Distilled, Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley, 2020.
- [2] Kennedy, Michael. *Product Development for the Lean Enterprise*. Oaklea

The information on this page is © 2010-2023 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs and contact us for permissions.

FRAMEWORK

- [Download SAFe](#)
- [Posters & Graphics](#)
- [Watch and download SAFe videos and presentations](#)
- [Blog](#)

TRAINING

- [Course Calendar](#)
- [About Certification](#)
- [Become a Trainer](#)

PARTNER

- [Becoming a Partner](#)
- [Partner Directory](#)

GET SOCIAL

- [Twitter](#)
- [Linkedin](#)
- [YouTube](#)

CONTENT & TRADEMARKS

- [FAQs on how to use SAFe content and trademarks](#)
- [Permissions Form](#)
- [Usage and Permissions](#)

SCALED AGILE, INC

CONTACT US

5400 Airport Blvd., Suite 300
Boulder, CO 80301 USA

BUSINESS HOURS

Weekdays: 9am to 5pm
Weekends: CLOSED

- [Privacy Policy](#)
- [Cookie Policy](#)
- [Your California Consumer Rights](#)

English