Glossary | SAFe® 4.0 for Lean Software and Systems Engineering

Agile Architecture

Agile architecture is a set of values and practices that support the active evolution of the design and architecture of a system, concurrent with the implementation of new business functionality. With this approach, the architecture of a system, even a large one, evolves over time while simultaneously supporting the needs of current users. This avoids Big Up-Front Design (BUFD) and the starting and stopping of stage-gated methods.

Agile Release Train

The Agile Release Train is a long-lived team of Agile Teams, typically consisting of 50 – 125 individuals. The ART aligns teams to a common mission and provides for a regular cadence for planning, development, and retrospective. Trains provide continuous product development flow and each train has the dedicated resources necessary to continuously define, build, and test valuable and evaluate-able capabilities every two weeks.

Agile Teams

The Agile Team is a cross-functional group of five to nine individuals who have the ability and authority to define, build, and test solution value—all in a short-iteration timebox. The team includes the individuals necessary to successfully deliver this value, supported by specialists where applicable.

Architectural Runway

Architectural runway provides one of the means by which SAFe implements the concepts of Agile architecture. This runway provides the necessary technical basis for developing business initiatives and implementing new features and capabilities. An architectural runway exists when the enterprise's platforms have sufficient existing technological infrastructure to support the implementation of the highest-priority, near-term features without excessive, delay-inducing redesign.

Budgets

SAFe provides strategies for Lean-Agile budgeting that funds value streams instead of projects. This empowers value streams with their own dedicated budget for rapid decision-making and flexible value delivery, while Program Portfolio Management (PPM) retains control of total spending, which is adjusted over time.

Built-in Quality

Built-in quality is one of the four core values of SAFe. The enterprise's ability to deliver new functionality with the fastest sustainable lead time, along with the ability to be able to react to rapidly changing business conditions, is dependent on built-in quality.

Business Epic

See Epic.

Business Owners

Business Owners are a small group of stakeholders (typically three to five) who have the ultimate fiduciary, governance, efficacy, and ROI responsibility for the value delivered by a specific release train. Business Owners typically have management responsibility for Customer relationships, development, solution quality, deployment, operations, Product Management, and architecture.

Capability

Capabilities are similar to features; however, they account for higher-level behaviors of the solution, which often spans multiple ARTs. Capabilities are maintained in the value stream backlog and are sized to fit in a program increment, so that each PI delivers solution value.

CapEx and OpEx

A value stream budget may include both CapEx and OpEx elements. CapEx typically captures the expenses required to purchase, upgrade, or fix tangible physical assets or other property used to support solution building. In some cases, CapEx may also capture elements of the costs of labor for development of certain intangible assets. OpEx costs typically include salaries and overhead, contract labor, materials, supplies, and other items directly related to solution development activities.

Communities of Practice

A Community of Practice (CoP) is an informal group of team members and other experts, acting within the context of a program or enterprise, that has a mission of sharing practical knowledge in one or more relevant domains.

Continuous Integration

Continuous integration is a practice whereby team members integrate and validate their work frequently; in the case of software, this can occur at least daily or even multiple times per day. Where possible, integration is verified by automated build and test environments that quickly identify integration problems and defects.

Coordination

See Value Stream Coordination.

Core Values

SAFe respects and reflects four core values: alignment, built-in quality, transparency, and program execution.

Customer

The Customer is whoever consumes the work of a value stream. They are the ultimate arbiters of value delivered. Whether internal or external to the development organization, they are an integral part of the development value stream.

Develop on Cadence

Basing routine development activities on a fast, synchronous cadence—a regular, predictive rhythm of important events—helps manage the inherent variability in systems development. This is a fundamental premise of SAFe. Its effects can be seen directly on the Big Picture, with the fast cadence of synchronized, short iterations followed by the further integration of those iterations into larger program increments.

DevOps

DevOps is a mindset, culture, and set of technical practices that stresses communication, collaboration, and close cooperation between Agile development teams and other technology professionals who are necessary for developing, testing, deploying, and maintaining software and systems.

Economic Framework

An economic framework is a set of decision rules that aligns everyone to the financial objectives of the mission, including budget considerations driven from the program portfolio. SAFe's first Lean-Agile principle is to take an economic view; the economic framework captures the essential economic elements for successful solution development.

Enabler Capability

Enabler capabilities occur at the value stream level, where they capture work of that type. As these enablers are a type of capability, they share the same attributes, including a statement of benefits and acceptance criteria, and they must be structured so as to fit within a single PI.

Enabler Epic

Enabler epics are a type of epic, and as such are written using the value statement format defined for epics. They tend to cut across value streams and PIs. They require a lightweight business case to support their implementation. They are identified and tracked through the portfolio Kanban system.

Enabler Feature

Enabler features occur at the program level, where they capture work of that type. As these enablers are a type of feature, they share the same attributes, including a statement of benefits and acceptance criteria, and are structured so as to fit within a single PI.

Enabler Story

Enabler stories, as a type of story, must fit in iterations. However, while they may not require user voice format, they have acceptance criteria to clarify their requirements and support demonstration and testing.

Enablers

Enablers encapsulate the exploration and the architectural and infrastructure development activities necessary to support some future solution capability. They occur at all levels of the framework and are described as enabler epics, enabler capabilities, enabler features, and enabler stories, depending on their level.

Enterprise

The enterprise represents the business entity that has the ultimate strategy, fiduciary, and governance authority for the value streams that constitute a SAFe portfolio. Each SAFe portfolio exists in the broader enterprise context; that is the source of the business strategy that the portfolio must address. The enterprise also provides the general governance model for all portfolios.

Enterprise Architect

The Enterprise Architect works with business stakeholders and Solution and System Architects to drive holistic technology implementation across value streams. The Enterprise Architect relies on continuous feedback, fosters adaptive design and engineering practices, and drives collaboration of programs and teams around a common technical vision.

Epic

Epics are significant initiatives that help guide value streams toward the larger aim of the portfolio. They are investment intensive and far ranging in impact. They require a formulation and analysis of cost, impact, and opportunity in a lightweight business case, as well as financial approval before implementation. There are two kinds of epics: business epics and enabler epics, and they may appear at the portfolio, value stream, and program levels.

Epic Owners

Epic Owners have the responsibility of shepherding epics through the portfolio Kanban system. They develop the business case and, when approved, work directly with the key stakeholders on the affected trains to help realize the implementation.

Feature

A Feature is a service provided by the system that fulfills stakeholder needs. Each is developed by a single Agile Release Train. They are maintained in the program backlog and are sized to fit in a program increment so that each PI delivers conceptual integrity. Each feature includes a statement of benefits and defined acceptance criteria..

Implementing 123

Implementing SAFe 1-2-3 is a proven, successful pattern for SAFe implementation. It describes three basic steps: (1) train implementers and Lean-Agile change agents; (2) train all executives, managers, and leaders; (3) train teams and launch Agile Release Trains.

Innovation and Planning Iteration

SAFe provides for periodic innovation and planning iterations, which serve a variety of purposes. They provide an estimating buffer for meeting objectives, a dedicated time for inspect and adapt and PI Planning activities, a cadence-based opportunity for innovation, time for continuing education, time for working on the technical infrastructure and other impediments, and time for backlog refinement.

Inspect and Adapt

Inspect and adapt (I&A) is a regular event, held at the end of each PI, that provides time to demonstrate the solution; get feedback; and then reflect, problem solve, and identify improvement actions. The improvement items can then be immediately incorporated into PI planning.

Iteration Execution

In SAFe, iterations have a fixed, two-week timebox. Agile Teams take items from the iteration backlog and define, build, and test them into the system baseline during this two-week period.

Iteration Goals

Iteration goals are a high-level summary of the business and technical goals that the team and Product Owner agree to accomplish in an iteration. In SAFe, iteration goals are integral to the effective coordination of an Agile Release Train as a self-organizing, self-managing team of teams.

Iteration Planning

Iteration planning is the ceremony during which all team members plan the upcoming iteration. The output of the meeting includes the iteration backlog, consisting of the stories and acceptance criteria committed to in the iteration; a statement of iteration goals; and a commitment by the team to the work needed to achieve those goals.

Iteration Retrospective

The iteration retrospective is a short team meeting held at the end of an iteration, wherein team members gather in a private, safe environment to discuss the efficacy of their practices and define improvements for the upcoming period.

Iterations

Iterations (sprints in Scrum) are a strict timebox in which teams deliver incremental value in the form of working, tested software and systems. Each iteration has a standard pattern: plan the iteration; commit to a goal; execute; demo the work to the key stakeholders; and, finally, hold a retrospective, wherein the team analyzes and determines actions necessary to improve their performance.

Lean-Agile Leaders

Lean-Agile Leaders are lifelong learners, managers, teachers, and coaches who help teams build better software systems through understanding and exhibiting the values, principles, and practices of Lean, systems thinking, and Agile software development. Lean-Agile Leaders adhere to the principles of Lean-Agile leadership.

Lean-Agile Mindset

By applying the Agile Manifesto and Lean thinking, a Lean-Agile mindset provides a comprehensive approach to Lean-Agile development that focuses on understanding and optimizing the flow of value from concept to delivery. SAFe's House of Lean is based on delivering value in the sustainably shortest lead time, respect for people and culture, flow, innovation, and relentless improvement—supported by a foundation of Lean-Agile leadership.

Metrics

The primary measure in SAFe is the objective measurement of working solutions. This is determined empirically, by demonstration throughout and at the end of every iteration and program increment. There are a number of additional intermediate and long-term measures as well, metrics that teams, programs, and portfolios can use to measure progress.

Milestones

Milestones mark specific progress points on the development timeline, and they can be invaluable in measuring and monitoring the progress and risk of a program. As opposed to phase-gate milestones, SAFe milestones are based on PIs, planned learning points, and fixed dates.

Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE) is the application of modeling and modeling tools to the requirements, design, analysis, and verification activities in solution development. MBSE provides a cost-effective way to learn about system characteristics prior to and during construction, and it helps manage the complexity and cost of large-system documentation.

Nonfunctional Requirements

Nonfunctional requirements describe system attributes such as security, reliability, performance, maintainability, scalability, and usability. They can also be constraints or restrictions on the design of the system.

PI Planning

PI planning is the seminal, cadence-based, face-to-face planning event that serves as the heartbeat of the Agile Release Train. It is integral and essential to SAFe.

Portfolio Backlog

The portfolio backlog is the highest-level backlog in SAFe. It provides a holding mechanism for the upcoming business and enabler epics required to create a portfolio solution set, a set that provides the competitive differentiation and operational efficiencies necessary to address the strategic themes and facilitate business success.

Portfolio Business Epic

Portfolio business epics capture the largest cross-cutting, business-facing initiatives that occur within a portfolio.

Portfolio Kanban

The SAFe portfolio Kanban system is used primarily to identify and manage the flow of epics that affect the course of action for value streams and the Agile Release Trains (ARTs) that realize them.

Portfolio Level

The SAFe portfolio level is the highest level of concern in SAFe. It provides the basic constructs for organizing the Lean-Agile enterprise around the flow of value via one or more value streams, each of which develops the systems and solutions necessary to meet the strategic intent.

Pre- and Post-PI Planning

The pre- and post-PI planning meetings allow ARTs and suppliers in large value streams to build an aligned plan for the next PI. The pre- and post-PI planning meetings serve as a wrapper for program level PI planning, where the actual, detailed planning takes place.

Product Management

Product Management is responsible for identifying Customer needs. They own the ART vision and roadmaps, pricing, licensing, ROI, and the program backlog. They drive PI objectives and release content via prioritized features and acceptance criteria, and they accept features into the baseline.

Product Owner

The Product Owner is the team member responsible for defining stories and prioritizing the team backlog. The Product Owner is also a member of the extended Product Manager/Product Owner team, understanding and contributing to the program backlog, vision, and roadmap.

Program Backlog

The program backlog is the single, definitive repository for all the upcoming work anticipated to advance the Agile Release Train solution. The backlog consists primarily of features intended to address user needs and deliver business benefits; it also includes the enabler features necessary to build the architectural runway.

Program Epics

Program epics are initiatives that are large enough to warrant analysis and a lightweight business case, but they are constrained to a single Agile Release Train. Unlike features, which are small enough to fit inside a single PI, program epics could take several PIs to develop. They can be a result of value stream or portfolio epics, or they may arise locally as ARTs reason about initiatives that represent larger effort and value.

Program Increment

A Program Increment (PI) is the larger development timebox that uses cadence and synchronization to facilitate planning, limit WIP, provide for aggregation of newsworthy value for feedback, and ensure consistent program level retrospectives. It is composed of multiple development iterations and an innovation and planning iteration. Due to its scope, the PI also provides the cadence for consideration of portfolio level considerations and roadmaps.

Program Kanban

The program Kanban helps ensure that features are analyzed prior to reaching an iteration boundary. They are estimated and prioritized appropriately, and feature acceptance criteria are established.

Program Level

The program level is where people and other resources are applied to some important, long-lived enterprise mission. Programs in SAFe are delivered by long-lived Agile Release Trains, which deliver a portion (in some cases all) of a value stream.

Program PI Objectives

The aggregation of each team’s PI objectives becomes the program PI objectives, which are approved and assigned business value by the Business Owners. If value stream level PI planning is needed, then the programs’ PI objectives are synthesized and aggregated to become value stream PI objectives.

Program Portfolio Management

Program Portfolio Management (PPM) represents the function that has the highest-level strategy and fiduciary decision-making responsibility in an enterprise portfolio. The PPM function has responsibility for strategy and investment funding, program management, and governance.

Release

The goal of Lean-Agile is frequent delivery of valuable, working, and fully tested solution increments. This is accomplished via a stream of releases, each of which has been validated and approved for final efficacy of use and is accompanied by the documentation necessary to ensure successful application.

Release Any Time

SAFe provides a separation of concerns that provides development teams with the cadence and synchronization tools they need to manage complexity and rapid change in their environment, while allowing for either synchronous (occurring on the PI boundary) or asynchronous (occurring any time) releases of value to the market.

Release Management

Release Management is a function that assists with planning, managing, and governing releases. This function has the authority and responsibility to help guide the value stream toward the business goals. It may contain dedicated individuals, or it may simply be a role that various Lean-Agile leaders play in the portfolio.

Release Train Engineer

The Release Train Engineer (RTE) facilitates Agile Release Train processes and execution. The RTE escalate impediments, helps manage risk, helps ensure value delivery, and drives continuous improvement.

Roadmap

The roadmap communicates planned Agile Release Train and value stream deliverables and milestones over a time line. The roadmap includes committed deliverables and visibility into the forecasted deliverables of the next few PIs. It is developed and updated by solution and product management as the vision and delivery strategy evolve.

SAFe Principles

SAFe is based on nine immutable, underlying Lean and Agile principles. These are the fundamental tenets, the basic truths and economic underpinnings that drive the roles and practices that make SAFe effective.

Scrum Master

The SAFe Scrum Master is a servant leader and coach for the Agile team. Primary responsibilities include ensuring that the process is being followed; educating the team in Scrum, XP, and SAFe; eliminating impediments; and fostering the environment for high-performing team dynamics, continuous flow, and relentless improvement.

ScrumXP

ScrumXP is a lightweight yet disciplined and productive process for cross-functional, self-organized teams to operate within the context of SAFe. A ScrumXP team consists of five to nine people, collocated wherever possible. ScrumXP teams use Scrum project management practices and XP-inspired technical practices, and they visualize and measure the flow of value.

Set-Based Design

Set-based design is a practice that maintains multiple requirements and design options for a longer period in the development cycle. Empirical data is used to narrow focus based on the emergent knowledge.

Shared Services

Shared Services represent specialty roles that are necessary for the success of an Agile Release Train or value stream but that cannot be dedicated full time to any specific train. These may include security specialists, information architects, DBAs, technical writers, quality assurance, IT operations personnel, and more.

Solution

A solution is either a final product delivered to the ultimate economic buyer or, alternately, a set of systems that enable an operational value stream within the organization.

Solution Architect/Engineering

SAFe Solution Architect/Engineering represents the individuals and teams who have the technical responsibility for the overall architectural and engineering design of the solution. They help align the value stream and Agile Release Trains to a common technological and architectural vision.

Solution Context

Solution context identifies critical aspects of the target solution environment and its impact on the usage, installation, operation, and support of the solution itself. It impacts development priorities and infrastructure, test environments, solution capabilities, features, and nonfunctional requirements. It also establishes focus for DevOps and similar deployment considerations.

Solution Demo

The solution demo is the apex event of the PI cycle for a value stream. The results of all the development efforts from multiple ARTs—along with the contributions from suppliers—are integrated, evaluated, and made visible to the Customers and other stakeholders. This demo provides a regular cadence for objective evaluation of the solution and for gathering stakeholder and Customer feedback.

Solution Intent

Solution intent represents the repository for storing, managing, and communicating knowledge of the solution that the system builders are developing, as well as technical information about how they are going to build it. It includes specifications, designs, and tests for the current state of the solution, as well intended changes. It can be realized in many forms, from documents, spreadsheets, and whiteboard sessions to formal requirements and modeling tools.

Solution Management

Solution Management is the value stream content authority. They have primary responsibility for development and prioritization of the value stream backlog. They work with Customers to understand their needs, create the vision and roadmap, define requirements, and guide work through the value stream Kanban.

Spanning Palette

The spanning palette is not an artifact per se; rather, it comprises various roles and artifacts that may be applicable at any level of the framework. It is used to customize the framework implementation to a specific context. For a more complete discussion, see Program Level.

Stories

Stories are the primary artifact used to define system behavior in Agile development. Stories are not requirements; they are short, simple descriptions of a small piece of desired functionality, usually told from the user’s perspective and written in the user’s language. Each story is intended to support implementation of a small, vertical slice of system functionality, supporting highly incremental development.

Strategic Themes

Strategic themes are specific, itemized business objectives that connect the SAFe portfolio to the enterprise business strategy. They provide business context for decision-making within the portfolio, affecting the economic framework and investments in value streams and ARTs. They serve as inputs to the budget, portfolio, solution, and program backlog decisions.

Supplier

Suppliers develop and deliver components and subsystems that help Lean-Agile organizations deliver value to their Customers. Suppliers possess unique and distinctively competent skills and solutions and are experts in their technology; they can provide a high leverage point for fast and economical delivery.

System Architect/Engineering

System Architect/Engineering aligns the ARTs to a common technological and architectural vision of the solution under development. They participate in defining the system and subsystems, validate technology assumptions, and evaluate alternatives. They support system development through providing, communicating, and evolving the larger technological and architectural view of the solution.

System Demo

The system demo is a primary mechanism for evaluating the full ART system and gaining feedback from the stakeholders. It occurs at the end of every iteration and provides an integrated, aggregate view of the new features that have been delivered by all the teams on the train in the most recent iteration. It provides the ART with a fact-based measure of current, system-level progress within the Program Increment.

System Team

The System Team is a special Agile Team on the ART or value stream (sometimes both) that is chartered to provide assistance in building and using the Agile development environment infrastructure, including continuous integration and test automation, integrating assets from Agile Teams, and performing end-to-end solution testing. They often participate in demonstrating solutions in the system demo.

Team Backlog

The team backlog represents the collection of all the things a team needs to do to advance their portion of the system. It contains user and enabler stories that originate from the program backlog, as well as stories that arise locally from the team's specific context.

Team Demo

The team demo is used to measure the team's progress by showing working stories to the Product Owner and other team members and stakeholders, and to get their feedback at the end of each iteration. Teams demonstrate every story, spike, refactor, and new nonfunctional requirement in this demo.

Team Kanban

Team Kanban is a method that facilitates the flow of value by visualizing work flow, establishing work-in-process limits, measuring throughput, and continuously improving the process. Kanban is particularly useful for System Teams, DevOps, and maintenance teams, and for other situations where a response mandate, fast-changing priorities, and lower value of planning lead them to this choice.

Team Level

The team level describes the organization, artifact, role, activities, and process model for the Agile Teams who power the Agile Release Train.

Team PI Objectives

Team PI objectives are a summarized description of the specific business and technical goals that an Agile Team intends to achieve in the upcoming PI. Their purpose is to validate understanding of business and technical intent; focus alignment on outcomes rather than on process or tactical concerns; and summarize data in a way that enhances communication, alignment, and visibility.

Test-First

Test-first is the practice of developing and testing a system in small increments, often with the development of the test itself preceding the development of the code or component. In this way, tests serve to elaborate and better define the intended system behavior before the system is built, thereby enhancing quality.

User Experience

While Agile Teams have responsibility for implementing the solution, including the user-facing elements, User Experience (UX) designers support a consistent user experience across the components and systems of the larger solution.

Value Stream Backlog

The value stream backlog is the definitive repository for all the upcoming work anticipated to advance the solution. The backlog consists of upcoming capabilities, which can span multiple ARTs, as well as enablers that advance learning and build the architectural runway.

Value Stream Coordination

Value stream coordination provides guidance for managing dependencies across value streams in a portfolio.

Value Stream Engineer

The Value Stream Engineer facilitates value stream processes and execution. He escalates impediments, manages risk, and helps ensure value delivery and continuous improvement.

Value Stream Epics

Value stream epics are initiatives that are large enough to warrant analysis and a lightweight business case (see Epic) but are constrained to a single value stream. Unlike capabilities that are defined to be small enough to fit inside a single program increment, value stream epics could take several PIs to develop. They can arise as a result of portfolio epics, or they may arise locally as value streams plan larger initiatives.

Value Stream Kanban

The value stream Kanban helps ensure that value stream epics and capabilities are reasoned about and analyzed prior to reaching a PI boundary, that they are prioritized appropriately, and that acceptance criteria have been established to guide a high-fidelity implementation.

Value Stream Level

The value stream level supports builders of large and complex solutions that typically require multiple ARTs, as well as the contributions of Suppliers. This level is most often used by enterprises that face the largest systems challenges, building large-scale, multidisciplinary software and cyber-physical systems.

Value Stream PI Objectives

During the post-PI planning meeting, ART objectives are further summarized at the value stream level and become value stream objectives. This is the top level of PI objectives in SAFe, and they communicate to stakeholders what the value stream, as a whole, will deliver in the upcoming PI.

Value Streams

Value streams are the primary SAFe construct for understanding, organizing, and delivering value. Each value stream is a long-lived series of steps that an enterprise uses to provide a continuous flow of value to a Customer. Value streams are realized by Agile Release Trains.

Vision

The Vision describes a future view of the solution to be developed, reflecting Customer and stakeholders needs as well as features and capabilities that are proposed to address those needs. It provides the larger, contextual overview and purpose of the solution under development. Vision appears on the spanning palette and can be applied at any level in the framework.

Weighted Shortest Job First

Weighted Shortest Job First (WSJF) is an economic model for prioritizing “jobs” based on product development flow. WSJF is calculated as the cost of delay divided by job duration. In SAFe, “jobs” are the epics, features, and capabilities that are developed by ARTs. There are three primary elements to the cost of delay: 1) user-business value, 2) time criticality, and 3) risk reduction-opportunity enablement value.

Guide to acronyms and abbreviations

ART	Agile Release Train
BO	Business Owner
BV	Business Value
BVIR	Big, Visual, Information Radiator
CFD	Cumulative Flow Diagram
CapEx	Capital Expenses
CI	Continuous Integration
CoD	Cost of Delay
CoP	Community of Practice
DoD	Definition of Done
DSU	Daily Stand-up
EA	Enterprise Architect
EO	Epic Owner
FW	Firmware
HW	Hardware
I&A	Inspect & Adapt
IP	Innovation and Planning (iteration)
MBSE	Model-Based Systems Engineering
NFR	Nonfunctional Requirements
OE	Opportunity Enablement
OpEx	Operating Expenses
PDCA	Plan, Do Check, Adjust
PI	Program Increment
PM	Product Manager
PM/PO	Product Manager / Product Owner
PO	Product Owner
PPM	Program Portfolio Management
ROAM	Resolved, Owned, Accepted, Mitigated
RR	Risk Reduction
RTE	Release Train Engineer
S4T	SAFe for Teams
SAFe	Scaled Agile Framework
SA	SAFe Agilist
SM	Scrum Master
SMART	Specific, Measurable, Achievable, Realistic, Time-bound
SoS	Scrum of Scrums
SP	SAFe Practitioner
SPC	SAFe Program Consultant
SW	Software
UX	User Experience
VS	Value Stream
VSE	Value Stream Engineer
WIP	Work in Progress
WSJF	Weighted Shortest Job First
XP	Extreme Programming