

What is SAFe?

Learn about SAFe and its principles and understand how it differs from other agile frameworks.



BY JESSICA PIIKILÄ

BROWSE TOPICS

- Agile manifesto
 - > Scrum
 - > Kanban
 - Lean portfolio management
 - OKRs
 - Long-term agile planning
 - **What is SAFe?**
 - Spotify model
 - Organizational agility with Scrum@Scale
 - Scaling agile with Rosetta Stone
 - Agile iron triangle
 - Using Improvement Kata to support lean
 - The Large-Scale Scrum (LeSS) framework
 - Beyond the basics whitepaper
 - > Software development
 - > Design
 - > The agile advantage
 - DevOps
 - > Agile Teams
 - > Agile tutorials
 - > About the Agile Coach
- [All articles](#)

The Scaled Agile Framework® (SAFe®) is a set of organization and workflow patterns for [implementing agile practices at enterprise scale](#). The framework is a body of knowledge that includes structured guidance on roles and responsibilities, how to plan and manage the work, and values to uphold.

SAFe promotes alignment, collaboration, and delivery across large numbers of agile teams. It was formed around three primary bodies of knowledge: agile software development, lean product development, and systems thinking.

As businesses grow in size, SAFe provides a structured approach for scaling agile. There are four configurations in SAFe to accommodate various levels of scale: Essential SAFe, Large Solution SAFe, Portfolio SAFe, and Full SAFe.

Dean Leffingwell and Drew Jemilo released SAFe in 2011 to help organizations design better systems and software that better meet customers' changing needs. At that time, teams used traditional project management processes to deliver software. But as the need to rapidly respond to changing market conditions increased, new frameworks emerged to help businesses improve solution delivery across their enterprises, and SAFe was born. Today, SAFe is one of the most popular scaled agile delivery frameworks, and SAFe's worldwide community of practitioners continue to evolve it.

JIRA ALIGN

[Scaled Agile Framework \(SAFe®\) Business Solutions](#)

[Learn more →](#)



WHITEPAPER

Beyond the basics of scaling agile

Learn the new rules to transformational agility.

[Download the whitepaper →](#)

Core principles and values

Core Values

SAFe's core values describe the culture that leadership needs to foster and how people should behave within that culture in order to effectively use the framework.

Alignment

SAFe requires that communication planning and

SAFe requires that companies put planning and reflection cadences in place at all levels of the organization. With these in place, everyone understands the current state of the business, the goals, and how everyone should move together to achieve those goals. By synchronizing people and activities regularly, all levels of the portfolio stay in alignment. Information flows both upward and downward in a timely fashion, unlike traditional top-down, command and control structures.

Built-in quality

In the SAFe framework, agility should never come at the cost of quality. SAFe requires teams at all levels to define what “done” means for each task or project and to bake quality development practices into every working agreement. According to SAFe, there are five key dimensions of built-in quality: flow, architecture and design quality, code quality, system quality, and release quality.

Transparency

SAFe encourages trust-building behavior, including planning work in smaller batch sizes so problems can be surfaced sooner, providing real-time visibility into backlog progress across levels, and [inspect and adapt](#) rituals.

Program execution

Program execution is the heart of SAFe and powers everything else in the framework. Teams and programs must be able to deliver quality, working software and business value on a regular basis.

Leadership

SAFe requires lean-agile leadership behavior because only leaders can change the system and create the environment necessary to embrace all of the core values.

SAFe Principles

The [Scaled Agile Framework’s principles](#) are meant to improve the company as a whole by inspiring lean-agile decision making across functional and organizational boundaries. The principles are intended to influence the decisions of not just leaders and managers, but of everyone in the organization and condition their mindset to shift from traditional waterfall thinking to lean-agile thinking, where practices like Lean Portfolio Management are applied.

Principle #1 Take an economic view

Inspired by the theories on product development flow from Donald Reinertsen's best selling books, achieving the shortest sustainable lead-time requires each individual in the decision-making chain to understand the economic implications of delays. Delivering early and often isn't always enough. According to SAFe, sequencing jobs for maximum benefit, understanding economic trade-offs, and operating within lean budgets are all responsibilities that need to be shared throughout the organization. Many of the concepts and tools are drawn from Reinertsen's theories on product development flow.

Principle #2 Apply systems thinking

SAFe encourages people using the framework to apply systems thinking to three key areas: the solution itself, the enterprise building the system, and the [value streams](#). Solutions can refer to products, services, or systems delivered to the Customer, whether they are internal or external to the Enterprise.

Large solutions have many interconnected component parts, so team members should have a higher-level perspective on how their part fits into the bigger picture. When thinking about the enterprise building the system, people following SAFe should consider the organization's people, management, and processes. So, if an organization is looking to optimize the way people work, it may need to eliminate silos, become cross-functional, and form new working agreements with suppliers and customers. Finally, the enterprise should clearly define how value flows from concept to cash in the solution development value streams. Leaders and management need to maximize the flow of value across functional and organizational boundaries.

Principle #3 Assume variability; preserve options

By default, designing systems and software is an uncertain exercise. This principle is addresses this by bringing in the concept of set based design, which calls for retaining multiple requirements and design options for a longer period in the development cycle. Set based design also relies on empirical data to narrow the focus on the final design option further in process.

Set-based design helps inform decision making during times of uncertainty by identifying the options and intended outcomes, much like a strategic bet. The

concept of integrating “learning milestones”, which refer to a deadline for decision, is instrumental to set-based design. The more teams learn over time, the more choices they can eliminate. The more choices they eliminate, the easier it is to identify the best path forward and produce the best possible outcome for customers.

Principle #4 Build incrementally with fast, integrated learning cycles

Similar to Principle #3, this principle addresses risk and uncertainty through learning milestones. It is not enough for each component part of the system to prove functional, the whole system must be considered to assess the feasibility of the current design choices. Integration points must be planned on a regular cadence to accelerate faster learning cycles. These integration points are an example of Walter A Shewhart's [plan-do-check-adjust](#) cycle, a framework for continuous quality improvement and mechanism for controlling the variability of development. Shewart's work and the work he inspired are often within SAFe.

Principle #5 Base milestones on objective evaluation of working systems

Demonstration of an actual working system provides a better basis for decision making than a requirements document or some other superficial evaluation of success. Including stakeholders in those feasibility decisions early on supports trust-building and systems thinking.

Principle #6 Visualize and limit Work in Process (WIP), reduce batch sizes, and manage queue lengths

Limiting work in process helps stakeholders see exactly how work is playing out.

The three elements of this principle represent the primary ways for maximizing throughput and accelerating value delivery - or in other words, implementing “flow”. As the saying goes “How do you eat an elephant? One bite at a time.”

When you apply this to software development, this means limiting the amount of overlapping work, the complexity of each item of work, and the total amount of work tackled at a given time. Small batch sizes allow for constant validation that work is headed in the right direction. And managing queue lengths ...

This principle seeks to offer guidance on optimizing for this for the best results

Principle #7 Apply cadence, synchronize with cross-domain planning

Agile teams naturally apply cadence through sprints or iterations. Creating a cadence for all possible matters reduces complexity, addresses uncertainty, builds muscle memory, enforces quality, and instills collaboration. Synchronizing these cadences enables the people and the activities to move like cogs in the wheel where learned information informs decisions and incremental planning.

Principle #8 Unlock the intrinsic motivation of knowledge workers

Inspired by influential management consultant Peter Drucker and author Daniel Pink, this principle is one of our favorites! It's about unleashing the potential of teams and helping leadership take the perspective of coaching and serving their teams over a command-and-control mindset.

Principle #9 Decentralize decision making

Reducing queue lengths and taking an economic approach by decentralizing decision making, gives teams the autonomy they need to get work done. Leaders should preserve their decision-making authority for topics of strategic importance and enable teams to make informed choices on everything else.

How does SAFe Work?

Organizations that are ready to implement SAFe usually have executive-level sponsorship, a strong purpose for change, and a foundation in Scrum.

Scaled Agile, Inc. provides a [SAFe implementation roadmap](#) that contains detailed steps on how to get started and set up the organization for widespread adoption across portfolios. The 12 steps for implementing SAFe include:

- 1 Reaching the tipping point
- 2 Train lean-agile change agents
- 3 Train executives, managers, and leaders
- 4 Create a lean-agile center of excellence
- 5 Identify value streams and ARTs (Agile Release Trains)

- 6 Create the implementation plan
- 7 Prepare for ART launch
- 8 Train teams and launch the ART
- 9 Coach the ART execution
- 10 Launch more ARTs and value streams
- 11 Extend to the portfolio
- 12 Sustain and improve

How does SAFe compare to other scaled agile frameworks?

Although Scaled Agile Framework® (SAFe®) is widely adopted across enterprises with large software development teams, other scaled agile frameworks have gained traction over time. All frameworks for scaling agile share five main components: inspiration from the 12 [Agile Manifesto](#) principles, cadence, synchronization, Scrum, and quality development practices. Understanding other frameworks' origins, core differences, and the conditions for their successful application can help organizations choose which framework best suits their needs.

Want more background on some of the top scaled agile frameworks? Check out the [Agile at scale overview page on the Agile Coach](#).

SAFe vs. Scrum@Scale

In Scrum@Scale (S@S), everyone is part of an interchangeable Scrum team. Depending on their goals, networks of Scrum teams come together to form an ecosystem. The purpose of S@S is to create a network of Scrum teams through a 'scale-free architecture,' meaning, basic Scrum roles and events are linearly scaled without introducing new process dynamics. For example, one Scrum of Scrum (SoS) may not be enough for a very complex product with 25 Scrum teams, so a Scrum of Scrum of Scrums (SoSoS) with a Scrum of Scrum of Scrums Master (SoSM) may be needed.

Although S@S is generally less prescriptive, it does offer one guiding question to help organizations determine whether they're ready to scale: If you add more people to the system does performance increase exponentially or does productivity suffer?

Similarly to SAFe, S@S does offer reference content online including an extensive Scrum@Scale guide which is increasing in popularity.

S@S is most successful when

- The technology stack is object-oriented (i.e. vertical user stories can actually be delivered in two weeks)
- An organization's feature teams have T-shaped skills, product-centric values, and minimum bureaucracy
- An agile or Agile Lifecycle Management (ALM) tool is not required until practices are second nature
- The executive team is willing to practice scrum and remove impediments for the organization

SAFe vs. Large-Scale Scrum (LeSS)

Large-Scale Scrum (LeSS) takes a minimalist approach to roles, structure, and artifacts. Where SAFe offers four configurations to accommodate teams of greater and greater size and with increasingly complex solutions, LeSS offers two configurations: LeSS for two to eight teams and LeSS Huge for more than eight teams. LeSS also differs in its stance that product owners should have complete content authority and strategic influence, where SAFe encourages a more democratic approach. And while in SAFe many factors inform strategy, LeSS places an emphasis on a customer-centric approach focused on paying customers.

Similar to S@S, LeSS scales from Scrum events, artifacts, and roles. And both SAFe and LeSS emphasize systems thinking, lean thinking, and similar guiding principles. LeSS, however, places a heavy focus on waste reduction across the organization, with the goal of continuous improvement.

LeSS is most successful when

- Scrum teams have mastered Scrum
- Leadership is willing to continuously restructure and experiment for the greater good
- There is alignment on the definition of the product
- There is alignment on the definition of done
- External coaches are working with organizational, team, and technical groups
- There are feature teams versus component teams with T-shaped skills

- The organization is willing to get rid of project management paradigm completely

SAFe vs. DA

Unlike the rest of the frameworks described, Disciplined Agile (DA) is a toolkit that enables organizations to decide what way of working makes the most sense to fit them. It offers lightweight agile governance which is rooted in Scrum and Kanban, along with transformation knowledge in areas like HR and finance, governance, DevOps, portfolio management, and more. DA involves situationally employing different levels of scale for each project and places an emphasis on decision-making enablement to help guide strategic direction.

DA is most successful when

- Organizations want to define their own scaled agile path(s)
- Organizations want to remain flexible across the enterprise
- Organizations want to preserve process and/or framework choices

SAFe vs Spotify

The Spotify “model” is a people-driven, autonomous set of practices that can be applied for coordinating agile teams. It was never intended to be a model or framework, but some businesses have adopted it as such. Spotify places an emphasis on self-organizing, cross-functional, and co-located teams called "squads" (the equivalent of a scrum team).

Comparatively, SAFe has no such stipulation on the co-location of teams, for PI planning it is encouraged.

Squads are organized into larger units called "tribes". Dependencies between squads are few and handled through Scrum of Scrums when they occur. Knowledge sharing is enabled through "chapters" and "guilds," informal groups organized based on skill sets and interests.

Compared to other examples, where online resources, training courses, and certifications are available, resources on the Spotify model are limited to a publicly available blog and other companions pieces developed by its pioneers and fans. The model is growing in popularity, so it's likely we'll see more on this in the future.

Spotify is most successful when

- Applying the ideas in your own business context
- The organizational culture focuses on learning, allowing for mistakes, and taking controlled risks
- Teams and products are “loosely coupled, closely aligned” to avoid dependency conflicts

SAFe 5.0

A central tenet of SAFe is that it continues to evolve in collaboration with its community of practitioners around the world. Most recently, Scaled Agile, Inc. launched the 5.0 version of SAFe. Key changes included the addition of a 10th principle, "Organize around value," and changing Step 12 from "Sustain and improve" to "Accelerate." But there's much more involved. Curious to learn more? Check out our [What's new and what's changed in SAFe 5.0](#) blog.

Conclusion

Frameworks like SAFe and the ones discussed above provide a viable option for helping businesses effectively scale agile within their organizations and achieve their business outcomes. But just as important are the tools you choose to help you amplify existing practices and realize the full benefits of those practices. Enter, Atlassian's Jira Align, an enterprise agile planning platform built for SAFe. Using Jira Align you can improve visibility, strategic alignment, and enterprise adaptability in order to accelerate your digital transformation.

[Learn more](#)



JESSICA PIIKKILA

Jessica Piikkila has been in the agile transformation business for over 10 years with deep roots in product management and agile coaching across multiple industries and capacities. The agile mindset has gotten her through tough personal challenges in life and fosters her love of learning. You will see her cracking the enterprise agility code with our customers.

Agile Topics

Agile project management
Scrum
Kanban
Design

Software development
Product management
Teams
Agile at scale
DevOps

Sign up for more agile articles and tutorials.

Email

email@example.com

[Subscribe](#)



Up Next
[Spotify model →](#)