

Agile at scale

Movin' on up: scaling agile in large organizations

BROWSE TOPICS

- Agile manifesto
- > Scrum
 - Lean portfolio management
- Okras
- 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

Software development teams have proven that implementing agile frameworks, like [scrum](#) and [kanban](#), lets them deliver solutions to customers faster, with more predictability, and gives them the ability to react quickly based on new information. Implementing agile at the individual team level is relatively easy - the benefits are clear and the resources are plenty (like this very microsite!).

But the real challenge is extending it across multiple teams in a large organization. In other words, [implementing agile at scale](#).

READ ON BELOW

Agile at scale articles

- 

ARTICLE

Managing an agile portfolio

How to make agile practices work across a large portfolio of many teams and lots of developers.
- 

ARTICLE

Lean Portfolio Management

Learn why traditional portfolio management practices no longer work, while examining the advantages of Lean Portfolio Management practices.
- 

ARTICLE

Okrs

Objectives are memorable, qualitative descriptions of what you want to achieve. Key results are a set of metrics that measure your progress towards the obj
- 

ARTICLE

Long-term agile planning - 8 steps to get started

Learn how to build great software using long-term agile planning.
- 

ARTICLE

What is SAFe?

The Scaled Agile Framework® (SAFe®) is a set of organization and workflow patterns for implementing agile practices at enterprise scale.
- 

ARTICLE

The iron triangle of planning

Take a cue from the decades-old iron triangle of planning and learn how balancing different variables can help agile software teams scale.

TUTORIAL

Learn how to use epics in Jira Software

Use these step by step instructions to learn how to use and create epics in Jira Software

[Try this tutorial →](#)

[CONTINUED]

Why are companies scaling agile?

Today, businesses need to be able to adapt, at enterprise scale, in order to stay competitive. The means to do so: responding to customers' evolving needs and delighting them in the process, providing flexible/customizable solutions, supporting teams of teams working on a unified front, shifting mindsets to place technology as a strategic enabler, and inspiring agile ways of working outside of software and IT teams.

But without a clear-cut plan or framework, it's increasingly harder for companies that are scaling to predict delivery, manage cross-team dependencies, and focus on the right business objectives. As a result, this often leads to a decline in customer satisfaction, loss of market share or revenue, and more.

All of this is prompting companies to invest heavily in agile –to either capture the benefits of scaling agile that their software teams might have seen or to remain competitive in today's market. But while large enterprises all might agree on the need to scale agile, how to do it and what it looks like is a completely different discussion.

So, what is agile at scale?



Agile at scale is the ability to drive agile at the team level, while applying the same sustainable principles, practices, and outcomes at other layers of the organization.

Scaling agile is a cultural transformation, where the business' people, practices, and tools are committed to improving collaboration and the organization's ability to execute against its strategy.

Ultimately, changes across these areas will help decentralize decision-making, create greater transparency and alignment around work, and increase speed to market, all while hard coding the values of agile into the DNA of the organization.

Where are you on your agile at scale journey?

We like to chart how far along an organization is in its agile-at-scale journey by looking at how teams and individuals are adopting agile practices.

Organizations at the beginning of their journey may only have pockets of people practicing agile, and

work may be dominated by traditional project management procedures focused on managing a project from conception through to delivery.

Organizations that are further along may have scaled agile practices in play (or even the use of a framework). This may prompt cross-functional teams to organize in a way that improves efficiency, keeps them laser-focused on the value they're delivering, and helps them navigate change by empowering them to make proactive decisions to help them meet their business' objectives.

No matter where you find yourself today, acknowledge and respect your position, and start from there.

Popular frameworks for scaling agile

There's no right way to scale agile. But many organizations have had great success evolving their processes, teams, and cultures using frameworks for scaling agile.

Here's a brief overview of the top scaled agile frameworks to explore:

SAFe

The [Scaled Agile Framework® \(SAFe®\)](#) is a set of organization and workflow patterns for implementing agile practices at enterprise scale. It was formed around three primary bodies of knowledge: agile software development, lean product development, and systems thinking. SAFe promotes alignment, collaboration, and delivery across large numbers of agile teams.

LeSS

[Large-Scale Scrum \(LeSS\)](#) is essentially regular scrum applied to large-scale development. LeSS is based on the idea that scaling frameworks should be minimalistic (i.e. include less rules, roles, and artifacts) to drive success. However, both LeSS and SAFe share some common patterns: Scrum at the team level, many teams sharing a backlog, collaborative planning across multiple teams, along with the general principles of pull and self-organization that any smaller agile team may be familiar with.

DA

Disciplined Agile (DA), previously referred to as Disciplined Agile Delivery (DAD), is a learning-oriented process decision framework for IT solution delivery. It provides a solid foundation from which to scale agile solution delivery within enterprise-class organizations. DA utilizes Scrum and Kanban, along with transformation knowledge in areas like HR and finance, governance, DevOps, portfolio management and more. DA is often considered more flexible and easier to scale than other methods.

Spotify

Spotify's approach wasn't meant to be a framework per se, but the organization's take on agile has organically emerged as one. The [Spotify model](#) is a people-driven, autonomous framework for scaling agile. It stresses the importance of culture and networks and provides an example for dealing with multiple teams in a product development organization.

Scrum@Scale (S@S)

[Scrum@Scale](#) is an extension of the Scrum framework. Scrum@Scale is generally adopted by organizations that have already implemented Scrum successfully at the team level and are looking to spread it throughout the organization. The main goal is to align growing organizations around one common and shared set of goals. Coordination is managed through a Scrum of Scrums, which is comprised of Scrum Masters from each team, and a MetaScrum, made up of product owners.

Understanding the difference between frameworks for scaling agile

Frameworks can add unnecessary process when they're applied without thought or intent. But codifying shared rituals, common roles, and guiding principles for scaling agile in your organization has clear benefits, especially when the organization is new to agile ways of working. Here's a helpful resource to guide your understanding of how each framework applies in these key areas.

	SAFe (Scaled Agile Framework)	LeSS (Large Scale Scrum) & LeSS Huge	Spotify	DA (Disciplined Agile)	Scrum@Scale
Long term planning and strategy	●	●	●	●	●
Multi agile teams	●	●	●	●	●
Team of Teams	Agile Release Train (ART)	Area	Tribes	●	Scrum of Scrums
PM/PO	●	●	●	●	●
Scrum Master / Agile Coach	●	●	●	●	●
Release Engineer / Group Manager	●	●	●	●	●
Agile practice (scrum, kanban, etc.)	●	●	●	●	●
Demo	●	●	●	●	●
Retros	●	●	●	●	●
Customer driven / value focused	●	●	●	●	●
Dependency management	●	●	●	●	●
Strategy transparency	●	●	●	●	●
Portfolio management	●	●	●	●	●
Release on demand	●	●	●	●	●
Risk Management	●	●	●	●	●
DevOps	●	●	●	●	●

As you can see, when examined closely, these frameworks incorporate common organizational patterns involved in scaling agile. Regardless of whether or not you adopt a framework, we believe that by adopting some or all of these core patterns, you can reap many of the benefits of scaling agile in your organization.

7 essential principles for practicing agile at scale

While we acknowledge there isn't any one-size-fits-all approach to agile at scale, there are seven essential principles for practicing agile at scale, which should be considered. These principles are "must-haves," meaning, it'll be near impossible to be successful without them.

Regardless of whether you plan to use a framework or which one you choose, consider the below as guiding principles for what can be borrowed or formalized in your own organization.

- 1 Defined roles and organizational structure changes
- 2 Customer-centric organization and development
- 3 Agile/Scrum practices and cadence
- 4 Adoption maturity (Take time to change)
- 5 Dependency improvements
- 6 Bottom up & top down buy-in (Actually change)
- 7 People, Lean, and systems thinking

For more details on these principles and what they look like in practice, jump to our "Beyond the basics of scaling agile: the new rules to transformational agility" whitepaper below.

So, where to begin?

Scaling agile is not easy, and won't happen overnight! Whether your organization goes all-in on a scaled agile framework, or implements a homegrown process, remember that "agile at scale" is not the end goal. The end goal is to effectively execute your strategy.

Keep trying new ideas and making incremental improvements with this aim in mind. And, don't forget that the tools you use to support your business can play an important part in scaling agile.

For a deeper dive into the topics we've covered on this page, check out our "Beyond the basics of scaling agile: the new rules to transformational agility" whitepaper. Packed full of helpful information and anecdotes from experienced agilists, it's definitely worth a read.



BEYOND THE BASICS OF SCALING AGILE

Ensure your organization's



Transform your organization agile practices are built to last

Download our whitepaper to learn the new rules to transformational agility.

[Download the whitepaper](#)

For more details on how Atlassian can support you on your agile journey, visit our [agile at scale solutions](#) page or [compare the features of our agile at scale products](#).



JOSH BERMAN

Josh is a Product Marketing Manager on Atlassian's agile at scale solutions team. When he's not writing about agile best practices you can find him hiking the greenbelt in Austin, Texas.



TUTORIAL

Learn how to use epics in Jira Software

Use these step by step instructions to learn how to use and create epics in Jira Software

[Read this tutorial →](#)



ARTICLE

Managing an agile portfolio

How to make agile practices work across a large portfolio of many teams and lots of developers.

[Read this article →](#)

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

[Subscribe](#)



Up Next
[Managing an agile portfolio →](#)