Scaling Agile - Practices that scale

18/11/2018 • 8 minutos para ler • Colaboradores

Neste artigo

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Enterprise organizations adopt Agile practices for a number of reasons. Prime among these include:

- Decrease time-to-market, accelerate product delivery
- Improve organizational effectiveness to manage changing priorities
- Enhance software quality and delivery predictability
- Improve project visibility and reduce project risk

As your organization grows, you'll want to scale your practices to remain agile and meet changing goals. To do that, consider these two guiding principles:

- **What does success look like** to you, your teams, and your organization? What's of most interest: On-time delivery? Product quality? Predictability? Customer satisfaction?
- **Return to first principles**, return to the principles and shared values enumerated in the Agile manifesto As noted by Ken Schwaber, one of the founders of Scrum:
  - "Values and principles scale, but practices are context sensitive."
  - "Keep the values, keep the principles, think for yourself. A core premise of Agile is that the people doing the work are the people who can best figure out how to do it."

Create rhythm and flow

By adopting a shared cadence and set of periodic communications, you create a constant flow of activity throughout the organization. Practices that help create rhythm and flow within larger organizations include:

- **Shared cadence**: Regular sprints and releases establish the rhythm of the business. Having all teams work to a shared cadence helps with all coordination and collaboration activities.
- **Sprint emails**: To keep the organization and all teams informed about the progress and plans of feature teams, each feature team can email a summary of their previous sprint results and current sprint plans.
- **Sprint demos**: A quick--2 to 3 minute--video that illustrates a new feature the team produced. Links to such videos can be included within sprint emails.
- **Showcase meetings**: To inform other teams and elicit feedback about software under development, teams showcase the work they've done. Conduct these meetings at regular intervals throughout the project lifecycle, and open them to all interested parties.
- **Bug summary emails**: To support insight into product quality and to encourage maintaining bug discipline, periodically share quality metrics with the organization. These metrics may include active bugs per feature team, bug trends, and bugs per engineer.

- **Coordination meetings**: Hold meetings that coordinate teams at either regular intervals or as often as needed to address overlapping goals, dependencies, and risks.

Enthusiastic customers

Engaging customers throughout your product lifecycle is a primary Agile principle. Empower each team to interact directly with customers on the feature sets they own.

- **Continuous feedback**: Build in customer feedback loops. These can take many forms:
  - **Customer voice**: Make it easy for customers to give feedback, add ideas, and vote on next generation features. This is often done through a dedicated website.
  - **Product feedback**: In-product feedback buttons are another way to solicit feedback about the product experience or specific features.
  - **Customer demos**: Regularly scheduled demos that solicit feedback from your customers can help shape next generation products and keep you on track to build applications your customers want to consume.
- **Early adopter programs**: Such programs should be developed with the idea that all teams may want to participate as some point. Early adopters gain access to early versions of working software which they then can provide feedback. Oftentimes, these programs work by turning select feature flags on for an early adopter list.
- **Data-driven decisions**: Find ways to instrument your product to obtain useful data and that can test various hypotheses. Help drive to an experiment-friendly culture that celebrates learning.

Improve project visibility

The more insight you and your teams have into the goal, vision, and progress of the work being done, the better enabled you'll be to reduce risks and manage dependencies.

- **Team structure**: No matter how large your organization gets, structuring your organization around small teams of 6 to 9 scales. Create vertical, autonomous feature teams grouped under portfolio management areas.
- **Work breakdown structure**: Breaking down large goals, features, or requirements into smaller ones remains a stable of project management. By breaking down work into similar-sized tasks, teams can make better estimates and identify risks and dependencies.
- **Consolidated views**: Use your online tracking tools to aggregate work to gain knowledge across teams. Build dashboards to show progress and trends.
- **Experience reviews**: These meetings, held before development begins on a feature, are used to educate leadership on scenarios and priorities, collect feedback, set expectations, and surface any cross-team issues about the feature.

Productive workforce

Some specific Agile practices that scale well and lead to happier, engaged, and productive employees include:

- **Embedded leadership**: Empower teams and leaders within the organization to self-organize and self-manage as much as possible. Team autonomy increases organizational agility team effectiveness. Ensure teams have the corporate sponsorship needed to succeed.
- **Daily stand-ups**: Or, Scrum meetings help keeps teams focused on what they need to do daily to maximize their ability to meet their sprint commitments. As organizations grow, they should consider staggering these meetings so that cross-team participation can occur as needed.

- Scrum of scrums: Daily standups of members from different Agile teams meet daily to report work completed, next steps, and issues or blocks occurring within their representative teams.
- **Team communications**: Provide and encourage teams to share their practices and guidance, which they and other teams can access through the corporate network. Common tools used for this purpose include team wikis, OneNotes, or markdown sites.
- **Collaboration**: Encourage informal team-to-team communications as well as collaboration within the team. Institutionalizing practices such as code reviews, design reviews, spec reviews not only increase team collaboration but help develop individual as well as overall corporate competence.

Organizational culture

You improve organizational effectiveness by attending to the culture you want to build. Culture changes occur when individuals, teams, and organizations adopt one or more continuous improvement practices. Several scalable Agile practices include:

- **Retrospectives**: By asking questions such as: "What went well?", "What should we do differently?", and "What should we stop doing?" help teams reflect on how they can improve on their processes and practices. Retrospectives help teams surface what is working well and what needs improvement. Retrospectives can be conducted anytime and anywhere. However, institutionalizing certain retrospectives at a regular cadence help institutionalize continuous improvement practices. For example:
  - **Sprint retrospectives** can help teams identify areas to improve at a regular cadence.
  - **Release retrospectives** can help organizations identify areas to improve communications and internal practices and fuel improvement for the next release.

  - **Operational reviews**: are typically held monthly and include representatives from a whole value stream. Spanning a portfolio of projects and other initiatives and using objective, quantitative data, design these retrospectives to provoke discussions about the dynamics affecting performance between teams.

    See the Agile Retrospective Resource Wiki for ideas, tips, and tools for planning and conducting retrospectives.

- **Improvement tracking board**: Good ideas to improve processes can arise from any one at any time. Capturing those ideas in order to discuss and decide on how to act on them in a timely manner is a key to support process improvement efforts.

  A white board provides any easy and visual means with which to capture ideas. Also, you can create an Improvement tracking team and capture ideas that you track on an electronic Kanban board.

- **Institutionalize sharing**: Sharing best practices and communicating ideas helps all teams within an organization grow and improve. Developing a culture of learning is key to supporting this and other continuous improvement activities. Some ideas to consider:
  - In-house wikis
  - In-house distribution lists
  - Hackathon weeks or 10% hack time

  - Internal Agile support team to support teams adopt Agile practices

    The Culture Game provides a good resource for Agile managers to help teams adopt Agile and share best practices.

- **Communities of practice**: Support internal common disciplines (e.g., DBAs, SW Architects, UX design)

Working software

> *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."*
> *"Working software is the primary measure of progress."*
> - Agile manifesto

As the amount of software, features, and complexity increase, you'll need to adopt practices that help you produce consumable solutions.

- **Feature flags**: Use feature flags to enable or disable access to different features. Provide support for turning features on to early adopters to get working feedback.
- **Release trains**: Provide another type of cadence to deliver one or more features. Feature teams understand the pre-planned schedule of pushing out new features, and plan accordingly. Release trains can correspond to the same sprint cadence establish for the organization, or occur at a different cadence. See Scaled Agile Framework for how to setup sprints and release trains.
- **Continuous integration**: Adopt processes that eliminate manual work and instead automate the flow of software through the test, build, and deploy cycles.
- **Internal Open Source**: Bring the value and ethos that's developed in the Open Source Software community to your internal development teams.

Related articles

In addition to the above practices, you'll find additional guidance around scaling your Agile tools in the following topics:

- Agile culture
- Add teams
- Portfolio management
- Visibility across teams
- Scaling Agile to large teams

Industry resources

- Agile manifesto
- Agile Alliance
- Scaled Agile Lean Development - The Principles

Practices that don't scale

- **Estimating large initiatives**: Part of waterfall project methods involved estimating resources and schedules. The larger the initiatives, the less likely these estimates were of any value. As projects grow, risks and unforeseen issues and impediments can arise, invalidating many estimates.
- **Velocity**: While team velocity can provide a useful metric for gaining insight into how much work each team can complete during a sprint cycle, you can't add team velocities to gain meaningful or useful metrics. Also, using velocity gained from a number of teams to reliably perform long range forecasts is problematic. Teams vary in the way they estimate their work, and those variations increase over time.
- **Top-down prescriptive solutions**: One size does not fit all, and one solution typically does not fit all teams. Supporting team autonomy means letting teams find their own solutions.