

10 Tips for Scaling Agile

By Mike McEwen, PMP, Dynanet Director of Software Solutions

When something works, you keep doing it. And when managers see success in one area, they naturally want to repeat the results in other areas. That's often the case when organizations start reaping the benefits of Scrum and other Agile software development techniques, which deliver incremental value to customers by assigning development tasks to one or more time-boxed iterations. At first, it sounds easy to "scale up" Agile. Can't we just create more Agile teams and let them work on different parts of the same product? However, how do you balance central control and coordination with the need for independent, self-organizing teams? What about dependencies between teams (e.g., a feature assigned to Team A depends on a service that Team B is still working on)? And, at what point, do you get diminishing returns or things spin out of control?

On the last question, you can scale up to 9 or 10 teams according to Ken Schwaber (one of the fathers of Scrum and a signer of the Agile Manifesto). If you want to exceed that number, he recommends either slowing down development or building application infrastructure (e.g., web services) that carefully defines the functionality each team (or group of teams) must deliver. There are some product companies that have more than 100 Agile teams but before scaling to that level, perhaps, the question should be "How many new features can our product users really handle right now?"

If you've asked that question and still need to scale, here are ten tips for making multiple Agile teams more successful:

1. All teams working on the same product should pull stories from a single backlog maintained by a single product owner. If you need multiple product owners, designate one person to promote a vision that everyone shares and unify the other local (or functional area-specific) product owners into a consistent product ownership team.
2. Establish a product/program/portfolio-level team (e.g., of architects, traveling experts, infrastructure/deployment/configuration managers, program director or chief ScrumMaster, etc.) to monitor product/release-level progress, set standards, and identify best practices and share them across all teams.
3. Adopt software tools that allow all teams to see the performance of other teams and roll-up individual team results into a product/release-level view.
4. Promote transparency, decentralized decision-making, and trust between teams as much as possible. Address conflicts and personnel issues quickly.
5. Give everyone the same training on methods, collaboration tools, and the shared technical environments. Ask all teams to agree on the same definitions of done and ready.
6. Establish joint ceremonies by asking representatives/observers from each team to attend product design and dependency/issue resolution meetings (e.g., a Scrum of Scrum held once or twice weekly). Also allow representatives from each team to attend all daily standups, sprint planning, and retrospectives of other teams. You may want to have a general sprint planning meeting as well as team-specific planning meetings. Consider establishing overall retrospectives or lessons learned meetings. Encourage direct communication between teams to resolve issues as soon as possible.
7. Synchronize iterations by having the teams start and end their sprints on the same day. All teams should collectively produce a single (potentially shippable) product increment at the end of each sprint.

8. Make sure that each team is cross-functional (i.e., has all the skills it needs to complete tasks). Avoid skill-specific teams of people required to complete development tasks.
9. Implement continuous integration by asking developers to check their code into a central repository and synchronize with the repository several times a day. Avoid team-specific code branches.
10. Perform integrated, end-to-end testing as early as possible to ensure the code that each team delivers interoperates with everyone else's.

Dynanet has successfully scaled Agile for the United States Food and Drug Administration and other customers. Adopting these tips will enable you to increase your Agile development capacity and deliver more features in less time to customers. For additional scaling frameworks and best practices, check out the Large Scale Scrum (LeSS) and Scaled Agile Framework (SAFe®) websites.

Dynanet Corporation, a minority-owned small business located in Elkridge, Maryland, has provided a range of information technology (IT) services to multiple Government agencies. Our focus on customer satisfaction and performance has established long-term relationships with the United States Air Force (USAF), Food & Drug Administration (FDA), the General Services Administration (GSA), the Office of Personnel Management (OPM), and the States of Maryland and Pennsylvania. During the last few decades, our executive leadership team has built Dynanet's dedicated workforce from 20 employees to over 125 key personnel. Dynanet is committed to employee engagement and maintains a retention rate of over 90%. Our managers are PMP certified and our employees maintain many sought after professional certifications, such as Certified Scrum Masters and ITIL certified. Additionally, Dynanet has been appraised at CMMI Maturity Level 3 (ML3) and our quality management processes have been audited and found to be in compliance with the ISO 9001:2008 quality standards. Further, our company has a Top-Secret facility clearance. For more information, please visit Dynanet's website at www.dynanetcorp.com.