

Agile Agile

Scaling Agile

[NK Shrivastava](#) and [Phillip George](#), [RefineM LLC](#)

Agile is a solution for delivering working increments on a timely basis. Used properly, it can be a powerful methodology to help satisfy customers, increasing their engagement in the project, and also enhancing a project team's delivery capability. To get these benefits, the teams need to be the right size. Typically, teams should not consist of more than 5-9 people, which presents problems when an organization needs to scale agile. In this article, general principles of scaling agile, as well as four different frameworks (Scaled Agile Framework, Disciplined Agile Delivery, Scrum at Scale, and Large-Scale Scrum) are examined, and guidelines are presented for how to choose a scaled solution.

Why do you need to scale agile? The primary reason is that teams are not always built to the needs of agile, so doing agile on projects with larger teams and greater complexity can be difficult. In addition, process complexity can make implementing an agile solution difficult, as well as stakeholder requirements. Rather than turning to another methodology, though, you can take advantage of frameworks for scaling agile. Several such frameworks exist, and in this article, we will look at four of them: Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD), Scrum at Scale, and Large-Scale Scrum (LeSS). We will discuss some of the particular terms of each methodology, as well as some of the common terms. We will also discuss the challenges of scaling agile. Through this article, you will learn more about scaling agile, including better understanding of each framework and suggestions for how to choose the right framework for your company.

General Principles of Scaling Agile

Mountain Goat Software lays out some general principles of scaling a team (Mountain Goat Software, 2015). First, projects do not scale by increasing the team size, but rather by increasing the number of teams and employing additional ways of organizing them. One should not expect to see an agile team larger than nine people, but rather a greater number of teams, and a “team of teams” to organize them (“Scrum Team,” 2015) One common ceremony for scaled agile approaches is the “scrum of scrums” where each team sends one representative to a meeting of all the teams, which may happen two or three times a week instead of the usual daily meeting. Other than this meeting, teams proceed as normal (“Scrum Team,” 2015).

Other than the “scrum of scrums” and the emphasis on more teams rather than larger teams, many frameworks have different approaches.

Scaled Agile Framework (SAFe)

Scaled Agile Framework, or SAFe, is a solution created by Dean Leffingwell (Pinna, 2013). It is currently on version 3.0. SAFe, like other solutions, is focused on the agile enterprise; it incorporates planning at the levels of team, program, and portfolio so that organizations can build a solution for the whole enterprise rather than one team or project. Another benefit of SAFe is that it is organized around value streams so that agile is focused on the key activities that deliver value.

At the team level, teams use Scrum and XP practices to define, build, and test working software every two weeks (Pinna, 2013). Teams consist of five to nine members. These members adopt many of the roles that one would expect to find in Agile: a product owner, developers, testers, and a ScrumMaster. Teams develop in cadence so that they can match up with one of the key features of SAFe, the Agile Release Train, which operates at the program level.

One of the key features of SAFe is the Agile Release Train. The release train is focused on a particular value stream that includes around 5-10 teams. The train delivers a Potentially Shippable Increment (PSI) every 10 weeks (5 iterations) (Pinna, 2013). At the program level, there are some unique roles. The Release Train Engineer manages teams on the release train as well as the logistics of each release, acting as what Pinna calls an “Uber-ScrumMaster.” The Product Manager role has deep expertise of the value stream, like a Product Owner would, and is a role that a Program Manager or Senior Project Manager could fulfill. Finally, the release management team helps manage each release to a customer, and includes members from various functions such as marketing, development, and quality (Pinna, 2013).

Another key set of roles at the program level is shared resources, particularly in the area of user experience (UX). Any specialized resource that is needed by every team on the release train is shared among each of the teams, including UX designers, security specialists, and database specialists. This level of organization ensures a greater consistency across teams. It helps teams who might otherwise run into issues from lacking necessary expertise on their own team, and also helps prevent specialized roles from becoming overloaded.

Finally, at the portfolio level, there are investment themes, which last 6-12 months and drive the budgeting process rather than budgeting happening at the project level (Pinna, 2013). To support the investment themes, there are two types of epic at the portfolio level: business epic, which is customer-facing, and architectural epic, which is focused on a technology solution; both of these share the goal of fulfilling the investment theme (Pinna, 2013). The presence of Agile Release Trains to deliver consistent PSIs across value streams helps teams work to complete their business and architectural epics, which in turn helps fulfill investment themes.

Several certifications are available for those who want to learn more about SAFe. These include the SAFe Agilist (SA) SAFe Practitioner (SP) and SAFe Product Manager / Product Owner (SPM/PO) for practitioners, and also Program Consultant (SPC) Program Consultant and Trainer (SPCT) for those who wish to teach others how to implement SAFe (Scaled Agile Academy, 2015). More information about each certification can be found at the [Scaled Agile Academy](http://www.refineM.com).

Disciplined Agile Delivery (DAD)

Disciplined Agile Delivery, or DAD, is a hybrid solution developed from a book by Scott Ambler and Mark Lines. It is a hybrid framework, combining Scrum, XP, Kanban, Lean Software Development, and many other methods, with the goal of providing a people-first solution to agile delivery (“Introduction to DAD,” 2015). In DAD, the primary roles are Team Lead, Product Owner, Team Member, and Architecture Owner, all of which form a team. Stakeholders are also primary roles. DAD defines primary roles as the roles which will always be present on a project (“Introduction to DAD”, 2015).

Many of those roles are familiar to project teams and follow the same responsibilities as one would expect on a project. The Architecture Owner is one unique role to DAD. This person is responsible for solution design decisions, and is usually a senior developer but does not have to be (“Roles in disciplined agile delivery”, 2012, 18 December) Secondary roles are introduced as needed to help scale the system, including specialists, domain experts, technical experts, independent testers, and integrators. These special roles help facilitate scaled solutions by stepping in when needed, and may go unfilled when not needed. Besides these roles, DAD supports a wide range of delivery strategies, including agile/basic, advanced/Lean, Lean continuous delivery, and exploratory “Lean startup” (“Introduction to DAD,” 2015)

Figure 1 depicts scaling factors for DAD (MacIsaac, 2012, July 17):

Figure 1. Disciplined Agile Delivery Scaling Factors



Scrum at Scale

Scrum at Scale, introduced in mid-2014, was founded by Jeff Sutherland and Ken Schwaber. They proposed a solution to scale Scrum based on its core principles so that any company can size it for any project (Scrum Inc., 2014, 6 December). They advocate a modular scaled approach for greater versatility, ease of incremental deployment, and greater adherence to the modular nature of Scrum.

Three dimensions of growing Scrum are defined in a presentation for Agile Orlando 2014 (Brown and Sutherland, 2014). The first is scale, or the number of coordinating teams and the complexity of projects. The second is distribution, or the number of different geographic locations. The third is saturation, or the degree to which agile has broken down “traditional silos” within a company. This factor is important because scaled agile solutions are more likely to succeed if agile has a foothold within a company already, rather than being a new initiative. Brown and Sutherland share examples of Scrum at Scale succeeding in many different organizational contexts, from small to large companies (Brown and Sutherland, 2014).

Scrum Inc. (2014, July) summarizes the Scrum at Scale process in a graphic, “Modular Framework for Scaling Scrum,” which includes the enterprise, business units, and teams. The enterprise starts with strategic vision, then moves to backlog prioritization. Once the backlog is prioritized, it is decomposed and releases are planned. Teams implement the releases, deploying the products and receiving feedback to refine the strategic vision. Business units work together to promote cross-team coordination and continuous improvement, as well as remove impediments, which makes sense given that there is more power to remove impediments at the higher level.

Jeff Sutherland, writing for LinkedIn, explains that Scrum at Scale is differentiated by its adaptability and focus on scaling principles (Sutherland, 2014, 15 December). The pattern library, another key feature, is a knowledge base of scaled solutions that others can use and build on, allowing companies to base their attempts on what other companies did. The modular approach and knowledge-sharing are two keys of Scrum at Scale.

Large-Scale Scrum (LeSS)

Large-Scale Scrum is adapted from the book *Practices for Scaling Lean and Agile Development*, by Craig Larman and Bas Vodde, and is another scaled solution specific to Scrum. It is defined as regular Scrum plus “a set of additional rules and the set of tips that we have seen work in large multi-team, multisite, and offshore agile development initiatives” (“Large-Scale Scrum is Scrum,” 2014). In other words, it is Scrum applied to large-scale development. The LeSS framework incorporates Lean thinking, continuous improvement, and systems thinking in its set of guidelines.

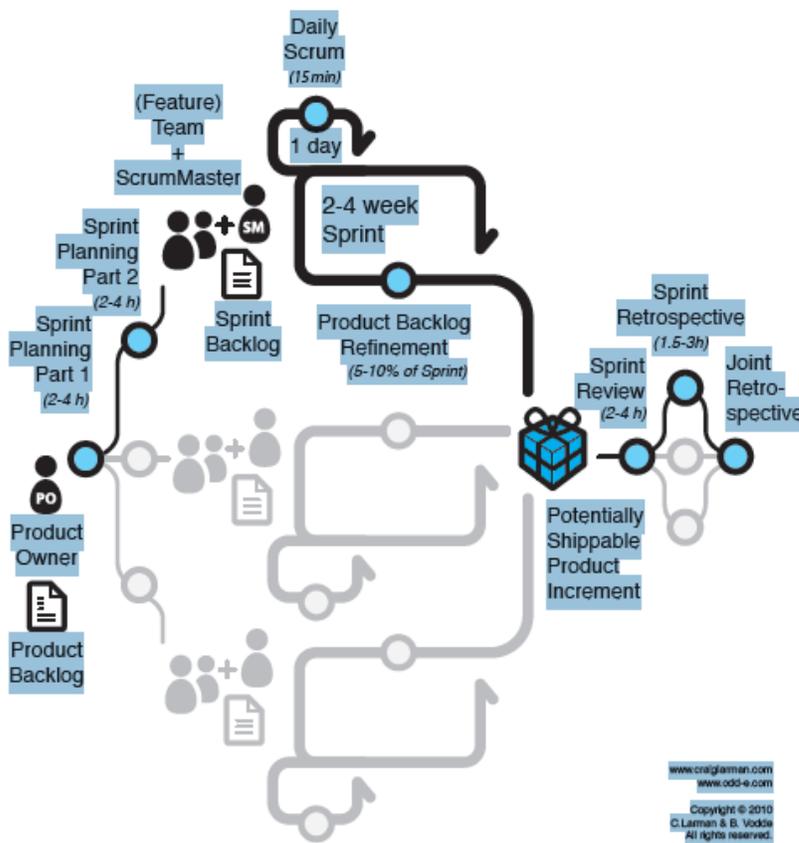
There are two levels of LeSS: regular LeSS and LeSS-Huge. Each level uses teams as the organizational building block. Regular LeSS is for 2-8 teams, while LeSS-Huge is for over 8 teams (“LeSS Rules,” 2014). In LeSS, ScrumMasters perform their role full-time for up to three teams at a time. Teams work on their sprints concurrently and share one Product Owner, Product Backlog, and Definition of Done for their shippable product. Each team has its own sprint backlog and retrospectives, with one common “Overall Retrospective” and sprint review (“LeSS Rules,” 2014). LeSS Huge adds “Requirement Areas” which are related clusters of customer requirements, each with its own product owner and group of 4-8 teams, which allows teams to focus on their particular area (“LeSS Rules,” 2014).

LeSS teams follow agile principles with some unique twists. In addition to being self-managed and cross-functional, they are also dedicated, which means that a team member spends 100 percent of his or her time on the same team. They are also co-located, and long-lived (rather than temporary) so they can develop stronger bonds and improve their performance (“Teams,” 2014).

Another key difference with LeSS is enforcing a reduced role for management. Under LeSS, the Product Owner decides what the team will do, and the team itself decides how they will do it; therefore, the role of management shifts to building capability. (“Role of Manager,” 2014). Building capability, in this context, is defined as managers removing impediments and developing a whole understanding of what teams are trying to do so they can coordinate with senior management to adapt the organization to teams’ goals.

LeSS presents itself as a scaled solution that works by focusing effort on what the customer wants and enforcing the principles of Scrum to help teams achieve it. Its key features, including the requirement areas, team makeup, and role of manager, support the idea that scaling successfully depends on first following the principles correctly. Figure 2 is a diagram provided by the [LeSS site](#) from *Practices for Scaling Lean and Agile Development*:

Figure 2. LeSS Diagram



Suggestions for Scaling Agile

The most important part of scaling agile is to determine what you are trying to accomplish by doing so. This knowledge will help you make an informed decision before proceeding. Blindly trying any of the approaches is likely to waste time and money. There can be a lot of overhead involved in implementing a method to scale agile, especially if people in your organization need to be trained and certified before implementing the solution.

The following table summarizes pros and cons of each solution:

Approach	Unique Features	Pros	Cons
Scaled Agile Framework (SAFe)	Agile Release Train, SAFe certifications	Organized around value streams, focuses on team, program, and portfolio, includes shared resources	Very prescriptive
Disciplined Agile Delivery (DAD)	Architecture Owner, secondary roles as needed	People-first system, highly flexible	May not have resources available to perform all roles needed
Scrum at Scale	Pattern library, modular approach	Promotes knowledge-sharing and incremental approaches to scaling	Not as effective if not using Scrum
Large-Scale Scrum (LeSS)	Requirement Areas, dedicated and long-lived teams	Focuses on pure Scrum and large-scale development	Not as effective if not using Scrum

Anjali Joshi of Salesforce.com, in describing Salesforce’s custom solution, Adaptive Delivery Methodology (ADM), presents five general tips for scaling agile in his presentation (Joshi, 2014, 29 September). The tips include focusing on values and culture, putting quality first, keeping the team engaged in continuous learning, and above all, having fun with the process. Joshi also advocates the “V2MOM approach,” which is described below, to focus thinking about the scaled solution:

- Vision: What impact will you have this year?
- Values: What is most important to you?
- Method: What actions are required to achieve the vision?
- Obstacles: What might stand in the way?
- Measures: What actual results are you aiming for?

Conclusion

Scaled Agile Framework, Scrum at Scale, and LeSS are all powerful ways to scale agile in order to see the positive effects of a scaled solution, which include greater ability for teams to achieve success on large, complex projects and to learn from each other as they go along. Being able to handle these projects means increased revenue and customer satisfaction capability. Whether you go with one of these solutions, or use the guidelines presented to create a custom solution, it is important to build a solution that works for you and your teams.

References

1. Brown, Alex and Sutherland, Jeff (2014). Scrum at scale: go modular for greater success. *Agile Orlando 2014*. Online Presentation. <http://www.scruminc.com/wp-content/uploads/2014/07/Scrum-at-Scale-A-Modular-Approach.pdf>
2. Disciplined Agile Delivery (2015). Introduction to DAD. Web. <https://disciplinedagiledelivery.wordpress.com/introduction-to-dad/>
3. Disciplined Agile Delivery (2012, 18 December), Roles in disciplined agile delivery. Web. <https://disciplinedagiledelivery.wordpress.com/2012/12/18/roles-in-disciplined-agile-delivery/>
4. Joshi, Anjali (2014, 29 September). Scrum at scale. *Slideshare.net*. Online Presentation. <http://www.slideshare.net/salesforceeng/scrumat-salesforcescrumalliancerio2014final>
5. LeSS Company B.V. (2014). Large scale scrum is scrum. *LeSS: More With Less*. Web. http://less.works/less/principles/large_scale_scrum_is_scrum.html
6. LeSS Company B.V. (2014). LeSS rules. *LeSS: More With Less*. Web. <http://less.works/less/rules/index.html>
7. LeSS Company B.V. (2014). Role of manager. *LeSS: More With Less*. Web. http://less.works/less/management/role_of_manager.html
8. LeSS Company B.V. (2014). Teams. *LeSS: More With Less*. Web. <http://less.works/less/structure/teams.html>
9. MacIsaac, Bruce (2012, 17 July). Disciplined agile delivery scaling factors. Web Image. <http://www.ibm.com/developerworks/rational/library/automate-software-development-processes/>
10. Mountain Goat Software (2015). The scrum team and scrum of scrums. Web. <http://www.mountaingoatsoftware.com/agile/scrum/team>
11. Pinna, Rob (2013). 41 things you need to know about the scaled agile framework (SAFe). *Rally Blogs*. Web. <https://www.rallydev.com/blog/agile/41-things-you-need-know-about-scaled-agile-framework-safe>
12. Scaled Agile Academy (2015). Certification overview. *Scaled Agile, Inc.* Web. <http://www.scaledagileacademy.com/?page=WhichCertification>
13. Scrum Inc. (2014, July). Modular framework for scaling scrum. Web Image. <http://www.scruminc.com/wp-content/uploads/2014/07/Scrum-EntInfographic-FINAL7-crop.jpg>
14. Sutherland, Jeff (2014, 15 December). Scrum at scale: changing the conversation. *LinkedIn Pulse*. Web. <https://www.linkedin.com/pulse/scrum-scale-changing-jeff-sutherland>