



Digital Business

Scaled and Distributed Scrum: Key Considerations for Success

These best practices in choosing a Scrum framework to handle large-scale Agile projects with distributed teams help avoid chaotic project situations while enabling today's digital companies to outpace traditional enterprises.

Executive Summary

Scrum, a framework for effective project team collaboration on a complex topic, is the most popular project approach in the Agile methodology. This lightweight framework, which is easy to understand but difficult to master, is used for work projects of all kinds, across business units with various sizes and organizational structures.

This has given rise to new challenges. Given its lightweight framework, Scrum's application in large project teams can lead to chaotic project situations. For this reason, many frameworks for scaling Scrum have been created over the last few years to accommodate a high number of project participants.

In addition, Scrum is no longer used only for co-located teams – teams are often spread across different offices,

countries or global regions. While the use of Scrum for large teams requires that the framework is adapted and extended, the distribution of project teams also leads to new challenges.

In regard to this, many companies that recently started working Agile are still looking for approaches to adapt their processes to those emerging challenges.

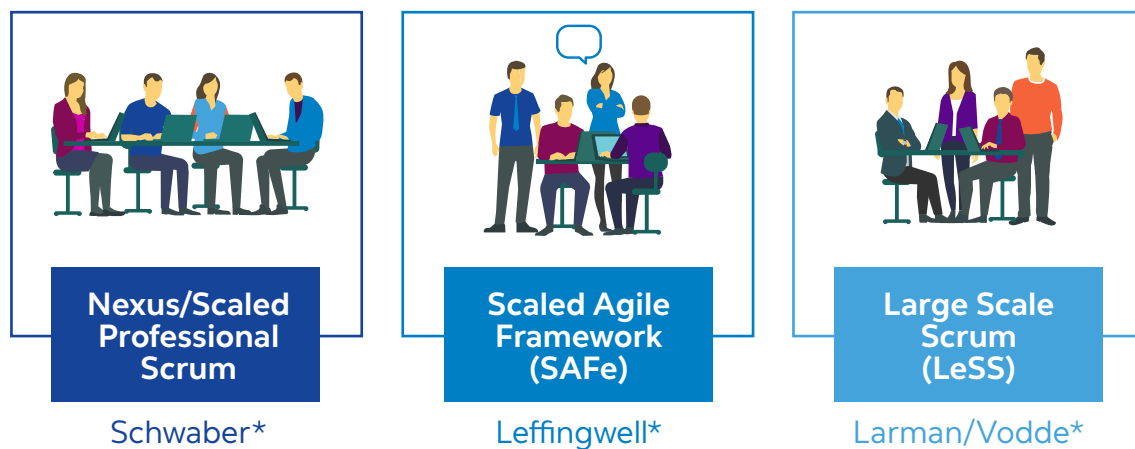
This white paper offers high-profile insights drawn from our client experiences where scaled and distributed Scrum techniques have been used in various constellations, and provides guidance on what additional actions can create even more value when Scrum is implemented for large and/or distributed projects.

Scaled Scrum: The frameworks

When Scrum is used for projects with more than two project teams (so-called “scaled Scrum”), it’s recommended to use an additional framework to deal with the increased dependencies and communication effort and to ensure that the project

still profits from the advantages of working Agile. Figure 1 provides an overview of three of the most frequently used Agile frameworks that deliver scaled Scrum.

The scaling frameworks



* Framework author/s
Figure 1

All such frameworks are built on the principles of Scrum, which pivots around self-organizing and cross-functional teams. As in traditional Scrum, teams in scaled Scrum also share one product backlog and plan the workload for Sprints collaboratively.¹

The differences are found in the structure of the frameworks, techniques used, popularity, flexibility, cost, support and proximity to the Scrum framework as a basis.

Choosing the right framework

When adapting a framework to scale a Scrum project, the first question is which framework to choose: A large variety of frameworks are available online and in the respective trade press and media. While some frameworks offer more detailed

guidance, others allow greater freedom and more flexible policy rules. However, this does not necessarily mean that one is better than another.

When choosing an appropriate solution, the team members involved and the needs of the organization are decisive. For example, a highly sophisticated and complex framework will not help anyone if the team has zero Agile experience yet and the project is starting in a few days. These are the important requirements and factors to consider when making a decision:

- I The company’s, and especially the project team’s, experience and skills:** Has the company used any frameworks already? Is there any existing internal support or expertise for a certain framework?

- I Support and training availability:** Is it possible to access support (if needed) and/or trainings for project participants? What kind of material is available to onboard team members?
- I Company culture:** Does the framework's philosophy fit your company's culture?
- I Available software and budget for new tools:** What is needed to establish the framework? Does the company have the tools to use it yet or can it buy them?
- I Regulations:** Are there regulatory requirements to fulfill concerning processes or documentation? Which supervisory authorities apply?

Furthermore, the implementation effort itself and the resulting structural changes play a major role and differ from framework to framework. The

big challenge here is to find the most suitable framework for the organization – according to its individual requirements.

The implementation effort and the level of guidance provided are the two main axes to differentiate the three frameworks (see Figure 2).

Nexus is a highly lightweight framework that only applies a few additional rules to Scrum. However, it is conceivable that some teams, particularly those who are new to Agile project management, will need more support than this framework provides. In this case, it makes sense to apply a more detailed framework, e.g. LeSS, with a medium level of detail, or SAFe, a more complex framework. The disadvantages of using a more highly detailed framework are that its implementation requires greater effort and it takes more time to internalize.

A matrix of frameworks

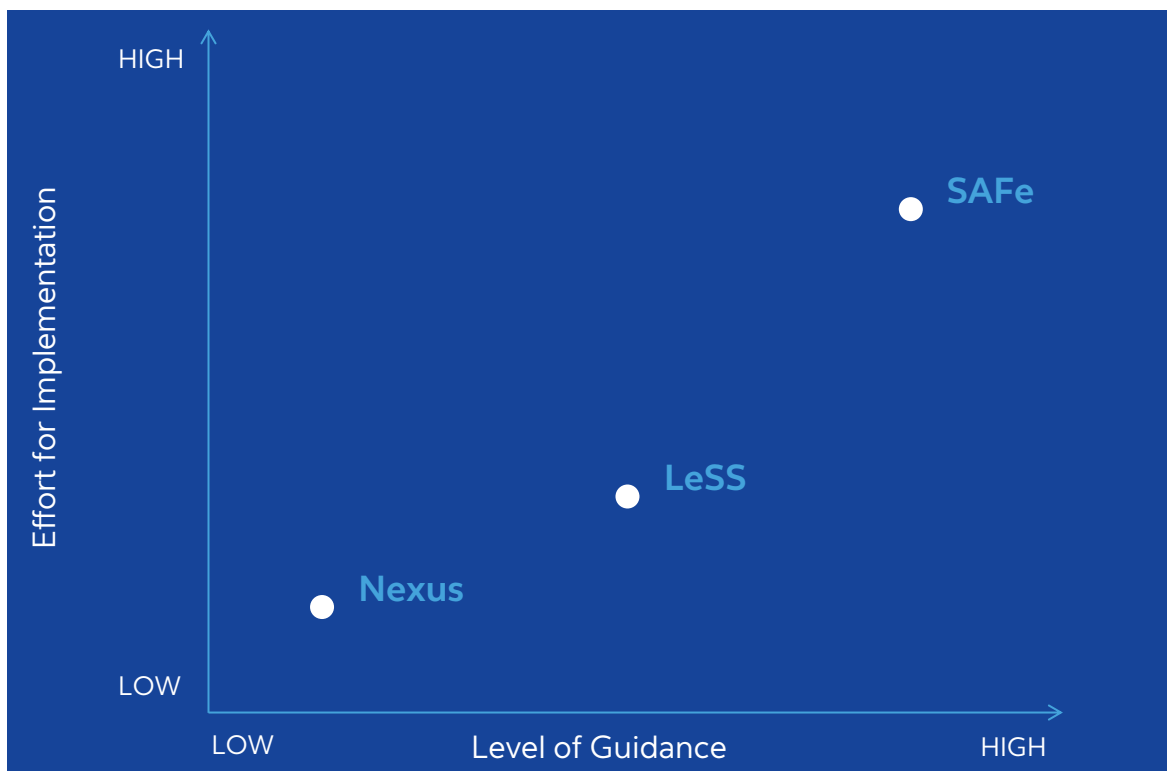


Figure 2

The main advantages of Nexus are its low learning and implementation barrier and that it does not create much overhead. On the other hand, this framework provides only moderate guidelines: teams need to align with additional custom rules and best practices to set up a well-functioning scaled Scrum environment.

Nexus

Nexus uses Scrum as its building block. Following the Nexus Guide, the framework binds and weaves together the work of three to nine Scrum teams working on a single product backlog to build an integrated increment to meet a project goal.² Its components are roles, events, artifacts and rules.

Nexus is an unrestricted framework that merely specifies some important add-ons to Scrum such as an additional integration team to coordinate, coach and supervise the Nexus implementation in order to ensure the best possible outcomes.

It is important to note that Nexus provides an exoskeleton for Scrum teams to work with, yet it is not a methodology that defines everything individuals need to know to scale Scrum. The main advantages of Nexus are its low learning and implementation barrier and that it does not create much overhead. On the other hand, this framework provides only moderate guidelines: teams need to align with additional custom rules and best practices to set up a well-functioning scaled Scrum environment.

Navigating the Nexus

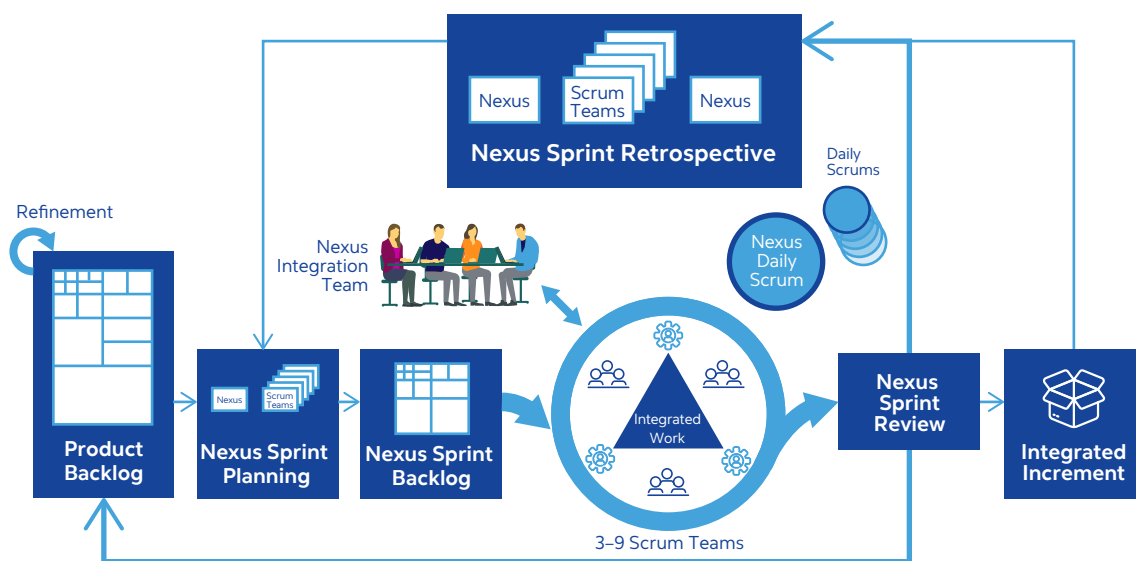


Figure 3

One example is the overview about the large variety of requirements: The more teams work on one project, the more requirements are transformed into a product increment every Sprint. The Nexus Guide states that the teams have to ensure that the work which is done forms a potentially shippable increment every Sprint (similar to basic Scrum). Also, the increased importance of dependencies between stories in a scaled Scrum environment is explained. However, the guide doesn't contain any information about how the needed overview can be achieved – the project teams have to either come up with their own ideas or take special trainings on Nexus. There are about 50+ best practices that can be learned from specialized Nexus trainers; these practices include instructions on how to properly do a story mapping. Otherwise, the large size of the product backlog can lead to double-entry accounting, various Excel lists and other unhelpful attempts to sort the backlog once the requirement chaos has developed.

Other aspects, such as organizational requirements that are dealt with in other frameworks (e.g., in LeSS), are not taken into account. Therefore, it is a major advantage if team members, especially those who are part of the Nexus integration team, already have experience in the use of (scaled) Scrum.

SAFe

SAFe is a framework of detailed scope, in which work is divided into value streams.³ Each value stream consists of work steps that are repeatedly applied to deliver value to the company, and these work steps are handled by a release train (five to 15 teams, up to 150 people).⁴

SAFe works with program increments (PIs), which consist of intervals of about 10 weeks. These are cadence-based and include other events such as the PI planning event where all teams meet for one or two days to arrange upcoming activities.

Playing it SAFe

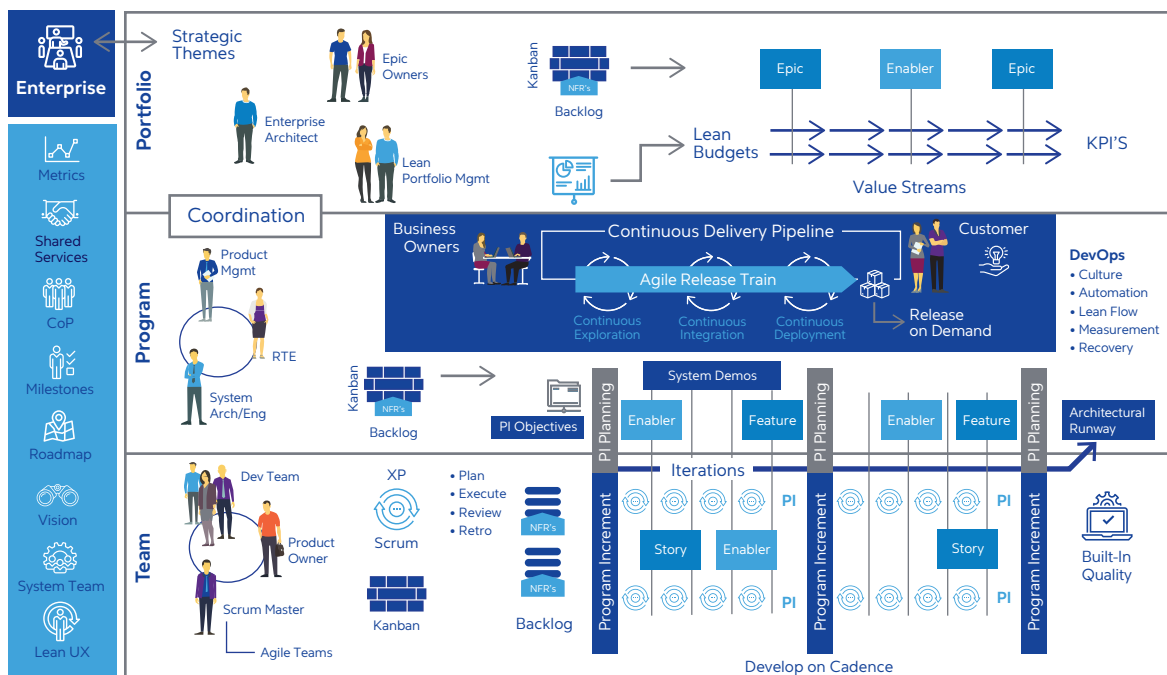


Figure 4

The framework consists of three levels: the portfolio level, where strategic decisions are made; the program level, where PIs are developed with the Agile Release Train; and the team level, where work for the single stories of the team's Sprint backlog is planned and performed⁵ (see Figure 4, previous page). The combination of these three levels equals one value stream. The SAFe framework itself can be scaled by adding additional value streams (large-scale SAFe).

Using the SAFe framework, the last Sprint of every PI (the Innovation & Planning Sprint) is blocked to finish work in progress, plan further increments and experiment with new ideas. Also, requirements in all other Sprints take up 30–70% of the Sprint to ensure sufficient time for defect management, user support and upgrading the development environment.

SAFe is not a purely Scrum-based framework, as it includes a Kanban process at the portfolio

level. It takes a lot of initial effort to understand and implement its structure properly, as it is a comparatively detailed framework. One advantage of SAFe is that it can natively be extended to cater to projects with more than 150 developers as it includes two scalable layers. In addition to scaling value streams on the program level, the team level can also be scaled by adding more teams to one Agile Release Train.

LeSS

LeSS is based on minimalism: It requires only a few process changes and adjustments to the basic Scrum framework to handle projects with multiple Scrum teams.⁶ The core principle of this framework is the understanding that teams should act as long-living feature teams. With the decrease of initial effort at the beginning of a new project, already established teams pick up work faster. Ideally, the teams are maintained on a cross-project basis, and they always have a single product owner and shared product backlog.

Where LeSS is more

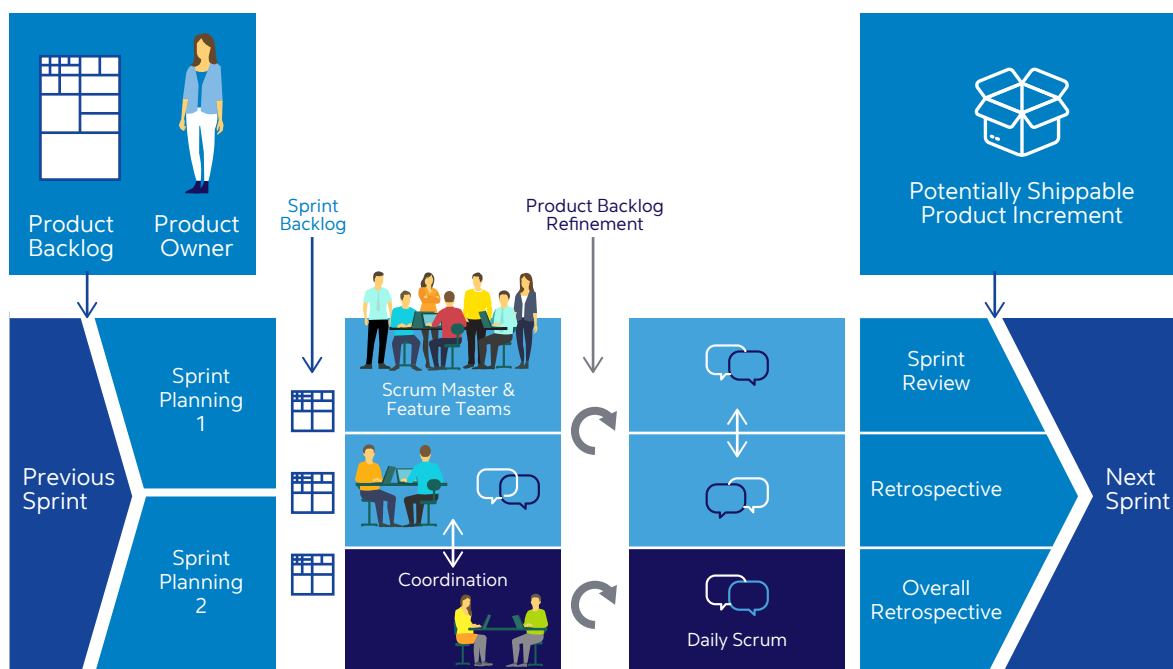


Figure 5

For teams which aim to use a framework with a certain level of detail and want to get their projects started with a minimum level of implementation effort, LeSS represents a good compromise. Since the framework offers a certain set of rules and also shows practical examples for customization, it offers strong guidance, yet doesn't add too much structure to Scrum. LeSS is optimized for three to eight teams; for larger projects, the LeSS Huge framework can be applied.⁷

LeSS's balanced level of guidance and ease make it a great choice for a large variety of project setups. For example, the LeSS approach aims to steer multiple teams in large projects with less effort while keeping processes as small and simple as possible.⁸ LeSS is based on Scrum with adaptations for scaled and distributed teams. At its most basic

level, it can be understood as a specification of how Scrum is meant to be applied.⁹ It applies the same principles and rules for its three to eight teams as Scrum with one team. With the involvement of more teams, a few amendments apply to ease communication and increase the overall efficiency of the project teams.¹⁰

At the core, LeSS is a set of principles drawn from practical experience that provides the foundation of the framework.¹¹

LeSS includes several rules that build the key elements of the framework to ease project setup and create a foundation. Furthermore, practical tips are given to enable project teams to adopt rules and principles to their underlying project case.

LeSS principles

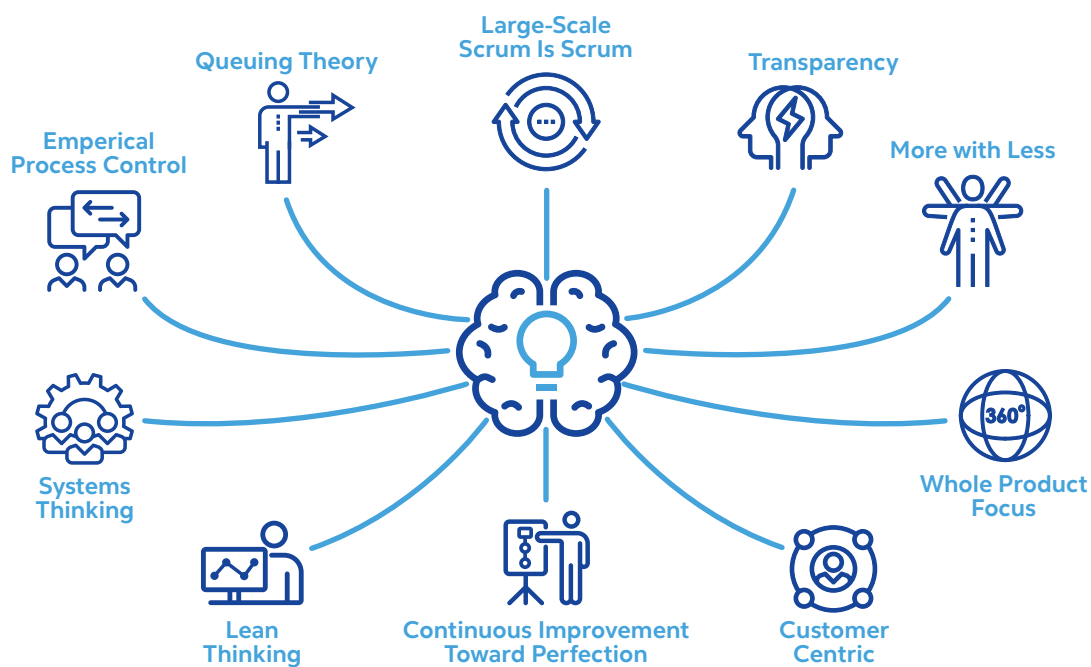


Figure 6

LeSS elements

The roles described within LeSS are similar to those found in Scrum, where only microinvasive changes apply: There is one product owner, three to eight feature teams and a Scrum master for every one to three teams.¹²

With LeSS, artifacts stay consistent; the only difference is that every team maintains its own Sprint backlog. But note that since the outcome of the project is one product, all teams share a single product backlog.

In scaled Scrum projects, product owners are at a high risk to lose oversight and for the product backlog to become overloaded when more than three teams work on the same product. The following events in the LeSS framework help facilitate coordination of teams and overcome project difficulties:

- Sprint planning part one includes the product owner, Scrum masters and some team representatives in order to coordinate the activities of different teams. Members discuss backlog items and then plan how to split up items for the upcoming Sprint.
- Sprint planning part two is held by each team separately, as in Scrum with one team. Within the meeting, team members discuss how to achieve the Sprint goal.
- A daily Scrum is held by each team independently.
- Overall product backlog refinement includes the product owner, subject matter experts, and either all members of the teams or representatives from each team. Participants analyze the requirements briefly. They estimate items and split those that are too big for estimation. Also, dependencies as well as strongly related items are identified. If a requirement will clearly be done by one team, it is forwarded to the respective team's product backlog refinement.

- Sprint review includes the product owner, all team members and important stakeholders. For better learning, a "science fair" approach may be considered.
- Overall retrospective includes the product owner, Scrum master and members from each team. It is used to identify and discuss challenges that affect more than one team.

LeSS Huge

LeSS Huge can be adopted for more than eight teams working on the same product. LeSS Huge is basically built of multiple LeSS frameworks stacked on top of each other, so there are multiple similarities LeSS and LeSS Huge share:

- One product owner who is responsible for the overall prioritization of the product backlog items and who decides on the scope and schedule of the releases.
- One product backlog, which contains all the requirements for the product to be developed.
- One definition of done for all teams: Even though it is possible that some of them expand the definition of done in their teams, the basis and minimum requirement for an increment to be considered potentially shippable is this shared definition.
- One shippable increment that is potentially releasable is the shared goal of all the teams.
- One Sprint for all teams, which starts and ends at the same time. This is viable for the creation of one shippable product increment per Sprint.

Differences apply regarding roles, artifacts and events due to increased complexity and the number of requirements and people. In addition to LeSS, LeSS Huge also comprises an area product owner, area backlog and Sprint execution per requirement area.



With LeSS Huge, teams are divided around major areas of customer focus, so-called requirement areas. This represents the principle of customer-centricity as introduced in LeSS.¹³ The complete project team decides on requirement areas such as management, performance, front end, etc. As soon as the basic structure is defined, items from the product backlog are classified into appropriate requirement areas. The product backlog is split into different area backlogs for each of those requirement areas.

Furthermore, a new role – the area product owner – is introduced. Every area product owner is responsible for one area backlog and for the product features it contains. The area product owner prioritizes backlog items and approaches development from a customer perspective. Nevertheless, the overall responsibility for the product backlog stays with the main product owner.

Every area feature team works within one requirement area and focuses on the items within one area product backlog. From a team's perspective, working in an area is similar to working in a smaller LeSS framework.¹⁴

The area product owner acts as product owner for teams working on the area backlog under his responsibility. When applying LeSS Huge, it is recommended to engage four to 10 teams per requirement area.¹⁵

Advantages of LeSS

If the organization has experience in working with Scrum, LeSS is readily adoptable as it is based on Scrum with only small adaptations. Applying LeSS offers further benefits to the entire organization and its projects:

- LeSS is centered on customers and the product.¹⁶ While other scaling frameworks divide teams according to single functions or architectural components, LeSS focuses on the customers' concerns.¹⁷
- LeSS concentrates on descaling the organization, enhancing simplification and setting up extended-duration team structures that remain intact beyond a single project period.¹⁸
- LeSS is a “no-brainer” framework for those organizations that are experienced in working with Scrum. It can easily be adopted and contains everything a project team will likely need to function.¹⁹
- The combination and coordination of multiple teams around one product is easier with LeSS than with other frameworks or no framework at all.
- LeSS creates synergies between projects and people, and the entire organization is included in the product value flow.²⁰

Scrum with distributed teams

Another decisive factor for the success of a Scrum project is the location of project participants. Working with decentralized teams is increasingly important as external teams who do not use Scrum usually work only with half of the velocity of central teams that use it.²¹ There are several ways to build teams relating to their location:

- I Co-located:** Teams are located at the same geographic place.
- I Nearshore:** The teams are located near each other – geographically as well as culturally. The distance is kept at a level where events from the Scrum cycle can be held at the same day/time and (except for the daily Scrum) in one place.
- I Far shore:** Teams are located abroad, travel time is long and both cultural and time differences apply.
- I Mixed:** A mix of the three aforementioned options (e.g., one team is co-located, five teams are placed nearshore). Having different locations within development teams themselves is also a possibility but is not recommended.

The three key advantages of near- or far shoring (both nuances of offshoring) are: lower labor cost, access to locally unavailable skills, and their flexibility in project size without layoffs.²² Thus, scaling with remote capacity enables the local team to remain stable as the project is scaled up or down.²³

Nevertheless, we came across some new challenges and “how to” questions arising when teams are not (fully) co-located while helping our clients introduce and adapt Agile concepts using scaled and distributed Scrum.

Shared vision

At the beginning of every project, especially in an Agile environment, the whole team needs to develop a shared understanding of the overall project vision and project objectives. A short period of co-location is therefore recommended to ensure the onboarding of all members. This ensures that teams work effectively and quickly get up to speed, and it lessens communication barriers as all team members have a shared understanding of the project. It can also be used to onboard team members with the LeSS framework and its principles. Based on our varied project experiences, we recommend that the LeSS advice “first educate everyone in depth” should be taken as seriously as possible – not only for the developers but also for the product owner, Scrum Master(s) and other stakeholders as the correct adaptation is vital for LeSS. This period of co-location also facilitates further knowledge transfer (e.g., on the domain) and helps establish shared working standards.²⁴



Way of work

Communication problems often develop due to differences in working style between onshore and offshore teams. Agile project management is sometimes understood and lived differently with regard to cultural differences. To counteract this, project teams should introduce working standards to which each individual commits. Those can be taken from respective frameworks and can additionally be defined by the teams. Also, management should support the team in finding a shared way of working, as this is seen as a major team achievement in distributed Scrum.²⁵ This can happen through LeSS framework-specific methodology like “go see” or “experiments over conformity” or via project-individual solutions. In one of our scaled Scrum projects, for example, the client stakeholders decided to really “go see” and spend some time at the developer’s sites abroad to have a look at how they work there and to also get a realistic sense of the environment, challenges and potential for optimization by supporting the teams. This diminishes the feeling of having a “black box” for the involved stakeholders.

A lack of cross-team learning can be an issue when it comes to scaled and decentralized teams. LeSS addresses this potential issue by hosting refinement and review meetings for co-located

and nearshore teams in large rooms, as known from science fairs. From our experience, even if it’s not possible to host all those Scrum events live, this get-together of all participants should happen every time the Sprint changes – and at least once a month. The time spent together, including common meetings, ensures that all team members and stakeholders have the chance to build trustworthy relationships and communicate directly about project progress or the difficulties that may affect development. In-person interaction can be crucial sometimes – and its lack can slow down the entire development process.

LeSS also provides a solution to handle dependency issues and the coordination between teams: It structures work-around features (LeSS) or requirement areas (LeSS Huge). Therefore, the overlap of requirements and tasks is kept at a minimum level.

Time zones

When setting up a distributed Scrum project, carefully consider the impact of multi-site development. The challenges for communication, knowledge-sharing and building reliable relationships are greater, plus distributed teams also often work in different time zones, where working

Agile project management is sometimes understood and lived differently with regard to cultural differences. To counteract this, project teams should introduce working standards to which each individual commits. Those can be taken from respective frameworks and can additionally be defined by the teams.

hours overlap only briefly. This challenge affects far-shore teams (at least partly). All larger Scrum meetings are attended by the complete team or at least by representatives from every team, which requires a high level of flexibility and understanding of working hours from all team members. Meeting times can be kept short by preparing the shared Scrum events to the maximum degree. However, this cannot be applied to the daily Scrum meeting, as the time difference would cause unease among team members. A proxy product owner role helps overcome these burdens by connecting and sharing information between two (or more) daily Scrum meetings held in different time zones.²⁶

Tools

Last but not least, applying distributed Scrum requires additional tools for conferencing, document sharing, etc. This becomes even more important than with co-located projects since team members do not have the option to meet for quick face-to-face conversations.²⁷ Furthermore, alternative digital solutions are required to maintain the common definition of done, discuss the list of

unsolved obstacles and overcome impediments.²⁸ Various project management tools, like JIRA or Confluence, provide further functionalities.

The LeSS framework attaches value to informal communication and networks. Team members are encouraged to use the easiest way to communicate – e.g., via code (using comments, for example), interdisciplinary meetings, mentoring for special components, open spaces, etc. In our experience, this leads to quick and easy communication patterns but can also create additional workload if there's not one source for already-made decisions. Therefore, we recommend the use of a common communication platform to meet challenges in team communication and to facilitate the exchange of information among teams. By applying this, all team members have access to the same information, can track changes and thus ensure a balanced distribution of information. Tools that can be used for this purpose reach from general project management tools (like Jira or Confluence, as mentioned above) to specific tools for distributed scrum.

If scaled Scrum is applied, the project organization must find a framework that meets the individual project requirements. From our experience supporting clients through the entire Agile journey, LeSS offers a good compromise in terms of detail and flexibility for most common project situations with a comparatively manageable implementation effort.



Looking forward

Both scaled and distributed Scrum can be used in different ways and come with specific challenges.

If scaled Scrum is applied, the project organization must find a framework that meets the individual project requirements. From our experience supporting clients through the entire Agile journey, LeSS offers a good compromise in terms of detail and flexibility for most common project situations with a comparatively manageable implementation effort.

On one hand, the framework provides defined rules, which offer guidance to project participants (if well adapted and trained); on the other hand, it also contains abstract principles that give each project the freedom to choose an appropriate setup, trying things out using the LeSS experiments while also avoiding common pitfalls. When using LeSS with our clients, we are well aware that LeSS does not offer everything needed for successful scaling, but it does form the basis for it – a starting point for inspection, adaption and improvement.

Furthermore, the location of the teams plays an important role. The use of offshore teams leads to (partly) lower labor costs and provides the

opportunity to apply know-how that is not locally available. It also increases flexibility in terms of project size. Yet, the offshore location causes challenges when it comes to developing a shared vision and working practices. Moreover, time zone differences add difficulties in scheduling meetings and require additional tools for planning and holding joint meetings.

LeSS supports distributed Scrum, as it works with feature teams who work independently to a certain degree. This reduces dependencies and overlaps within the teams. LeSS also promotes the stability of teams beyond a single project period. If these principles are lived by organizations, the effort of setting up new projects is significantly less and finding a common ground of work practices is simplified – even if teams are spread globally. Against this background, applying the LeSS framework for decentralized Scrum projects (nearshore, far shore or mixed) should be seen as a best practice.

Endnotes

- ¹ H. Brattberg, J. Grape, T. Björkholm (2017), “Doing Scrum with Multiple Teams: Comparing Scaling Frameworks,” www.infoq.com/articles/scrum-multiple-teams-frameworks.
- ² K. Schwaber (2018), Nexus Guide, p. 3.
- ³ D. Leffingwell (2017), Agile Release Train, www.scaledagileframework.com/agile-release-train.
- ⁴ Op. cit. Brattberg et al.
- ⁵ Op. cit. Leffingwell.
- ⁶ Op. cit. Brattberg et al.
- ⁷ Op. cit. Brattberg et al.
- ⁸ LeSS Works (2018c), Introduction to LeSS, LeSS Framework Summary, <https://less.works/less/framework/introduction.html#LeSSFrameworkSummary>.
- ⁹ Op. cit. Brattberg et al.
- ¹⁰ LeSS Works (2018a), LeSS Framework, <https://less.works/less/framework/introduction.html#LeSSFramework>.
- ¹¹ Ibid.
- ¹² Ibid.
- ¹³ LeSS Works (2018b), LeSS Huge Framework, <https://less.works/less/framework/introduction.html#LeSSHugeFramework>.
- ¹⁴ Ibid.
- ¹⁵ Ibid.
- ¹⁶ S. Powers (2015), “Why Large Scale Scrum,” www.adventureswithagile.com/2015/08/18/why-large-scale-scrum/.
- ¹⁷ Op. cit. LeSS Works, 2018b.
- ¹⁸ Op. cit. LeSS Works, 2018a.
- ¹⁹ Op. cit. Brattberg et al., p. 1.
- ²⁰ Op. cit. S.Powers.
- ²¹ J. Sutherland, A. Viktorov, et al. (2007), Distributed Scrum: Agile Project Management with Outsourced Development Teams, p. 3.
- ²² J. Sutherland, G. Schoonheim et al. (2009): Fully Distributed Scrum, Linear Scalability of Production between San Francisco and India.
- ²³ J. Sutherland, G. Schoonheim, M. Rijk (2009): Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams.
- ²⁴ Op. cit. J. Sutherland, G. Schoonheim et al., p. 3.
- ²⁵ S. Berczuk (2007), Back to Basics: The Role of Agile Principles in Success with a Distributed Scrum Team.
- ²⁶ Op. cit. J. Sutherland, G. Schoonheim et al., p. 3.
- ²⁷ Op. cit. S. Berczuk, p. 1.
- ²⁸ Op. cit. S. Berczuk, p. 3.

About the authors

Thorsten Schenk

Head of Digital Strategy Consulting, Cognizant Digital Business Germany

Thorsten Schenk has been the Head of Digital Strategy Consulting at Cognizant Digital Business Germany for the past two years. Before joining Cognizant, he worked as a CMO at a top 10 German digital bank for five years, spearheading the transformation of digital CX, marketing and sales. Thorsten brings 20 years of experience in digital transformation, 12 years of leadership in this area and five years of new digital methodology experience (Agile, design thinking, Lean start-up) which is – together with change management – one of his current focus areas. He can be reached at Thorsten.Schenk@cognizant.com | www.linkedin.com/in/thorsten-schenk-39163372/.

Annika Hajek

Manager, Digital Strategy Consulting, Cognizant Digital Business Germany

Annika Hajek has been a Manager at Cognizant Digital Business Germany since 2017. She has more than five years of experience within digital transformation, with an emphasis on digital and mobile strategy, user experience and change management. At Cognizant, Annika helps business leaders envision, build and run the business of the future using digital methodology like Agile project management and the MVP approach. Before Cognizant, she was program lead at a digital bank for the Agile end-to-end development of a new mobile checking account. Annika can be reached at Annika.Hajek@cognizant.com | de.linkedin.com/in/annika-hajek-3b686411a.

Cristina Rimari

Junior Consultant, Digital Strategy Consulting, Cognizant Digital Business Germany

Cristina Rimari joined Cognizant in January 2018. She has a background in information systems and management, with previous experience including project management and web-based application development at large companies in the aerospace and energy industries. Furthermore, her expertise lies in Agile development and requirement engineering. Cristina can be reached at Cristina.Rimari@cognizant.com | www.linkedin.com/in/cristina-rimari-149789128.

Kristina Schradstetter

Consultant, Digital Strategy Consulting, Cognizant Digital Business Germany

Kristina Schradstetter joined Cognizant as a Digital Strategy Consultant in early 2018. Before then, she had worked three years in the fashion sector and guided the digital transformation of a multinational SME from the North American market and managed web and online experience across European markets. Kristina brings experience in CX, digital marketing strategy and data analytics. She has a master's degree in strategic management. Kristina can be reached at Kristina.Schradstetter@cognizant.com | www.linkedin.com/in/kristina-schradstetter-10b300aa.

About Cognizant Digital Strategy Consulting

Every day the world shifts more toward new digital media and interactive experiences. With rapidly evolving technologies, changing consumer preferences and oftentimes competing channels, many organizations struggle with how to transform internally to meet the challenges of this new, always connected digital world. Cognizant Digital Strategy Consulting enables organizations to create engaging and consistent digital experiences across every touch point, providing new opportunities for growth.

About Cognizant

Cognizant (Nasdaq-100: CTSI) is one of the world's leading professional services companies, transforming clients' business, operating and technology models for the digital era. Our unique industry-based, consultative approach helps clients envision, build and run more innovative and efficient businesses. Headquartered in the U.S., Cognizant is ranked 194 on the Fortune 500 and is consistently listed among the most admired companies in the world. Learn how Cognizant helps clients lead with digital at www.cognizant.com or follow us @Cognizant.



World Headquarters

500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277

European Headquarters

1 Kingdom Street
Paddington Central
London W2 6BD England
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102

India Operations Headquarters

#5/535 Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060

APAC Headquarters

1 Changi Business Park Crescent,
Plaza 8@CBP # 07-04/05/06,
Tower A, Singapore 486025
Phone: + 65 6812 4051
Fax: + 65 6324 4051