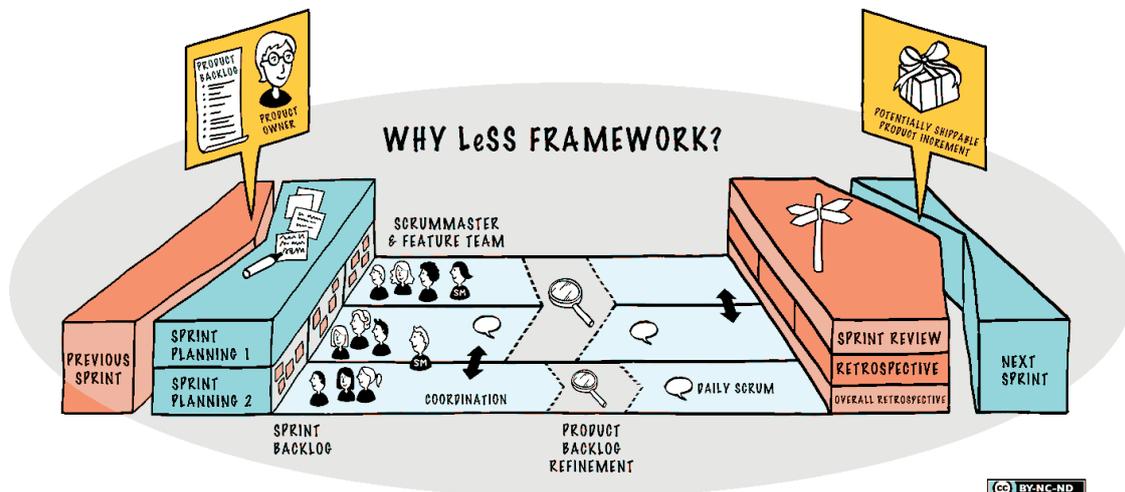


LeSS Framework



Scaling Scrum starts with understanding standard one-team Scrum. From that point, your organization must be able to understand and adopt LeSS, which requires examining the purpose of one-team Scrum elements and figuring out how to reach the same purpose while staying within the constraints of the standard Scrum rules.

Agile development with Scrum requires a deep organizational change to become agile. Therefore, neither Scrum nor LeSS should be considered as merely a practice. Rather, they form an organizational design framework.

Two Agile Scaling Frameworks

LeSS provides two different large-scale Scrum frameworks. Most of the scaling elements of LeSS are focused on directing the attention of all of the teams onto the whole product instead of “my part.” Global and “end-to-end” focus are perhaps the dominant problems to solve in scaling. The two frameworks – which are basically single-team Scrum scaled up – are:

- LeSS: Up to eight teams (of eight people each).
- LeSS Huge: Up to a few thousand people on one product.

What does it mean to be the same as One-Team Scrum?

LeSS is a scaled up version of one-team Scrum, and it maintains many of the practices and ideas of one-team Scrum. In LeSS, you will find:

- a single Product Backlog (because it's for a product, not a team),
- one Definition of Done for all teams,
- one Potentially Shippable Product Increment at the end of each Sprint,
- one Product Owner,
- many complete, cross-functional teams (with no single-specialist teams),
- one Sprint.

In LeSS all Teams are in a common Sprint to deliver a common shippable product, every Sprint.

What's Different in LeSS?

- **Sprint Planning Part 1:** In addition to the one Product Owner, it includes people from all teams. Let team members self-manage to decide their division of Product Backlog Items. Team members also discuss opportunities to find shared work and cooperate, especially for related items.
- **Sprint Planning Part 2:** This is held independently (and usually in parallel) by each Team, though sometimes for simple coordination and learning two or more Teams may hold it in the same room (in different areas).
- **Daily Scrum:** This is also held independently by each Team, though a member of Team A may observe Team B's Daily Scrum, to increase information sharing.
- **Coordination:** Just Talk, Communicate in Code, Travelers, Open Space, and Communities.
- **Overall PBR:** There may be an optional and short overall Product Backlog Refinement (PBR) meeting that includes the one Product Owner and people from all teams. The key purpose is to decide which teams are likely to implement which items and therefore select those items for later in-depth single-team PBR. It is also a chance to increase alignment with the Product Owner and all teams.
- **Product Backlog Refinement:** The only requirement in LeSS is single-team PBR, the same as in one-team Scrum. But a common and useful variation is multi-team PBR, where two or more Teams are in the same room together, to increase learning and coordination.
- **Sprint Review:** In addition to the one Product Owner, it includes people from all teams, and relevant customers/users and other stakeholders. For the phase of inspecting the product increment and new items, consider a "bazaar" or "science fair" style: a large room with multiple areas, each staffed by team members, where the items developed by teams are shown and discussed.
- **Overall Retrospective:** This is a new meeting not found in one-team Scrum, and its purpose is to explore improving the overall system, rather than focusing on one Team. The maximum duration is 45 minutes per week of Sprint. It includes the Product Owner, Scrum Masters, and rotating representatives from each Team.

Why LeSS?

Traditional sequential-lifecycle development doesn't work well. It doesn't work well for either small or large product development efforts. Since 2001, Agile development and Scrum in particular have revolutionized software development, but when asked how to apply Agile development to large groups many people say "don't" or "just use a small team" or "do Scrum at the team level." None of those answers is particularly useful, and while it is true that it is best to avoid adding people to your development effort, it is also true that large scale product development isn't going away so we need to discover ways to do it well.

We (Craig Larman and Bas Vodde) have been involved in software development for a long time in all sorts of roles in traditional sequential-lifecycle development, Unified Process, CMMi and others. None felt *right*. Scrum, on the other hand, felt right for single-team development. So, the question then became "How can we scale Scrum *without* losing its strength?"

LeSS is Scrum scaled

What is the strength of Scrum? That is not an easy question to answer. Of course, the concepts and principles behind Scrum, such as Transparency, Empirical Process Control, Iterative development, and Self-managing teams are critical. Those principles have been around for quite a while, however, so their inclusion does not explain Scrum's success. After much discussion, we have concluded:

Scrum hits the sweet spot between abstract principles and concrete practices.

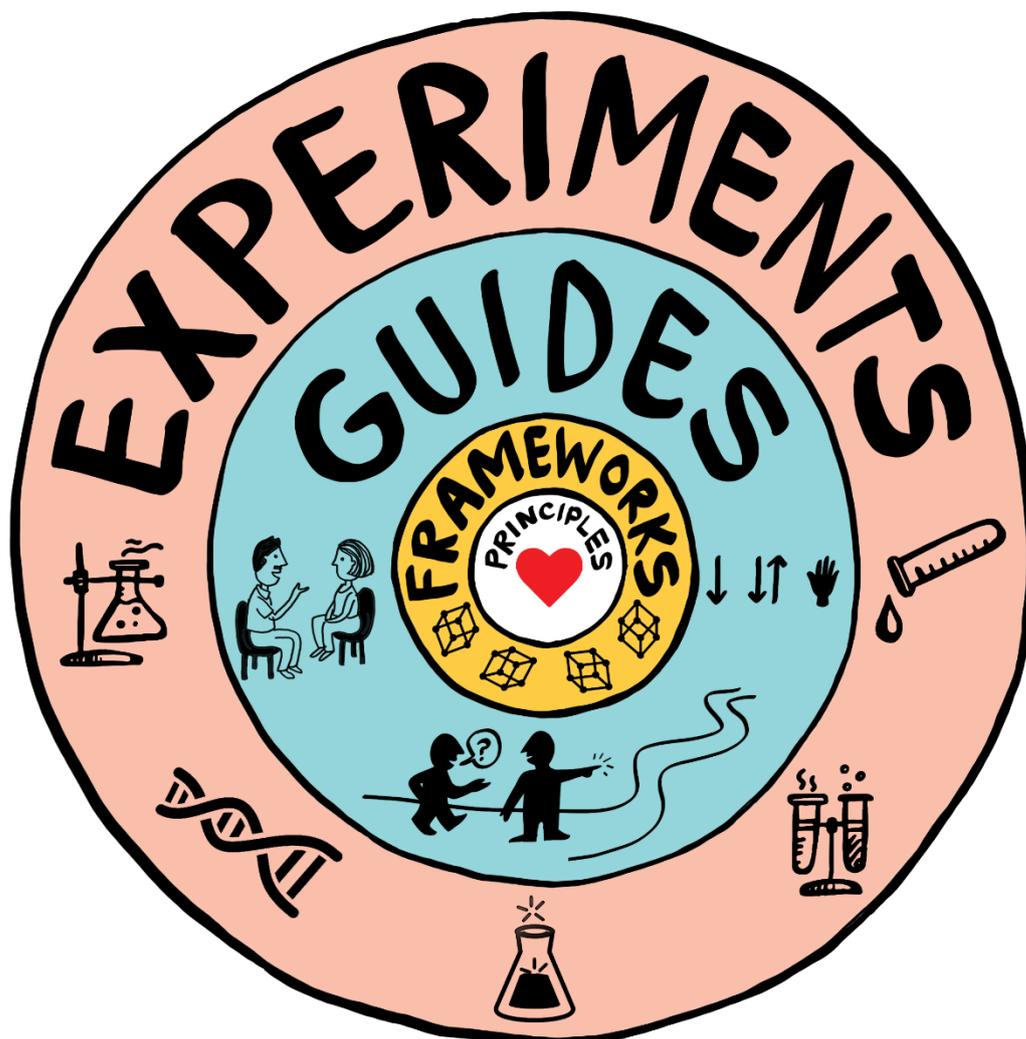
Thus, in order to keep Large-scale Scrum as Scrum, we'll need to find a similar balance, so that we will be able to say:

For large groups, LeSS hits the sweet spot between defined concrete elements and empirical process control.

This leads to some decisions:

- **LeSS needs to be simple**
When scaling, there is a tendency to add roles, artifacts, processes, etc. This should be avoided so that a process can empirically be created by the product group. Most other scaling frameworks fall into the trap of providing a defined process. In LeSS we want to avoid that trap.
- **LeSS is Scrum Scaled**
Rather than having Scrum as a building block for a scaled framework, we need to look at Scrum and for each element ask "Why is it there?" followed by "If we have more than one team, how can we achieve the same purpose on a larger scale?"
- **Scaled up instead of tailored down**
A common concept for process development is to define a universal, overarching framework and then contextually tailor it down. This does not work well because people often assume everything is needed in their particular context. This assumption often leads to the creation of bloated processes. LeSS should be kept to the minimum. We acknowledge that scaling will require "more" but instead of "polluting" LeSS with optional elements, we have separated out a different framework LeSS Huge.

The LeSS complete picture



LeSS as a framework

LeSS is more than a set of principles and experiments. It also provides a framework with rules. The LeSS Rules define what is LeSS (and what isn't) and they provide a concrete framework for applying LeSS. Within the LeSS Framework, product groups can apply the experiments and discover what works best for them at a certain moment.

There are no such things as best practices. There are only practices that are good within a certain context.

Principles Overview

Large-Scale Scrum is Scrum — It is not “new and improved Scrum.” LeSS is about applying the principles, elements, and purpose of Scrum in a large-scale context. Multiple-team Scrum, not multiple Scrum teams.

Empirical process control — Inspection and adaptation of the product, processes, organizational design, and practices to craft a situational appropriate organization based on Scrum, rather than following a detailed formula. And empirical process control requires and creates transparency.

Transparency — Based on tangible ‘done’ items, short cycles, working together, common definitions, and driving out fear in the workplace.

More with less — (1) In empirical process control: more learning with less defined processes. (2) In lean thinking: more value with less waste and overhead. (3) In scaling, more ownership, purpose, and joy with less roles, artifacts, and special groups.

Whole-product focus — One Product Backlog, one Product Owner, one potentially shippable product increment, one Sprint — regardless if there are 3 or 33 teams. Customers want the product, not a part.

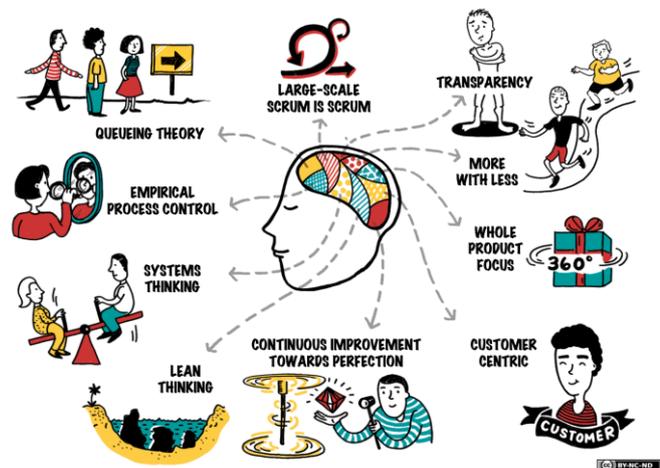
Customer-centric — Identify value and waste in the eyes of the paying customer. Reduce the cycle time from their perspective. Increase feedback loops with real customers. Everyone understands how their work today directly relates to paying customers.

Continuous improvement towards perfection — Create and deliver a product all the time, without defects, that utterly delights customers, improves the environment, and makes lives better. Do humble and radical improvement experiments each Sprint towards that.

Systems thinking — See, understand, and optimize the whole system (not parts), and explore system dynamics. Avoid the local and sub-optimizations of focusing on the ‘efficiency’ or ‘productivity’ of individuals and individual teams. Customers care about the overall concept-to-cash cycle time and flow, not individual steps.

Lean thinking — Create an organizational system whose foundation is managers-as-teachers who apply and teach systems thinking and lean thinking, manage to improve, and who practice **Go See** at **gemba**. Add the two pillars of respect for people and continuous improvement. All towards the goal of **perfection**.

Queueing theory — Understand how systems with queues behave in the R&D domain, and apply those insights to managing queue sizes, work-in-progress limits, multitasking, work packages, and variability.



LeSS Rules (April 2018)

LeSS Framework Rules

The LeSS framework applies to products with 2-“8” teams.

LeSS Structure

- Structure the organization using real teams as the basic organizational building block.
- Each team is (1) self-managing, (2) cross-functional, (3) co-located, and (4) long-lived.
- The majority of the teams are customer-focused feature teams.
- Scrum Masters are responsible for a well-working LeSS adoption. Their focus is towards the Teams, Product Owner, organization, and development practices. A Scrum Master does not focus on just one team but on the overall organizational system.
- A Scrum Master is a dedicated full-time role.
- One Scrum Master can serve 1-3 teams.
- In LeSS, managers are optional, but if managers do exist their role is likely to change. Their focus shifts from managing the day-to-day product work to improving the value-delivering capability of the product development system.
- Managers’ role is to improve the product development system by practicing [Go See](#), encouraging Stop & Fix, and “experiments over conformance”.
- For the product group, establish the complete LeSS structure “at the start”; this is vital for a LeSS adoption.
- For the larger organization beyond the product group, adopt LeSS evolutionarily using Go and See to create an organization where experimentation and improvement is the norm.

LeSS Product

- There is one Product Owner and one Product Backlog for the complete shippable product.
- The Product Owner shouldn’t work alone on Product Backlog refinement; he is supported by the multiple Teams working directly with customers/users and other stakeholders.
- All prioritization goes through the Product Owner, but clarification is as much as possible directly between the Teams and customer/users and other stakeholders.
- The definition of product should be as broad and end-user/customer centric as is practical. Over time, the definition of product might expand. Broader definitions are preferred.
- One Definition of Done for the whole product common for all teams.
- Each team can have their own stronger Definition of Done by expanding the common one.
- The perfection goal is to improve the Definition of Done so that it results in a shippable product each Sprint (or even more frequently).

LeSS Sprint

- There is one product-level Sprint, not a different Sprint for each Team. Each Team starts and ends the Sprint at the same time. Each Sprint results in an integrated whole product.
- Sprint Planning consists of two parts: Sprint Planning One is common for all teams while Sprint Planning Two is usually done separately for each team. Do multi-team Sprint Planning Two in a shared space for closely related items.
- Sprint Planning One is attended by the Product Owner and Teams or Team representatives. They together tentatively select the items that each team will work on that Sprint. The Teams identify opportunities to work together and final questions are clarified.
- Each Team has their own Sprint Backlog.
- Sprint Planning Two is for Teams to decide how they will do the selected items. This usually involves design and the creation of their Sprint Backlogs.
- Each Team has their own Daily Scrum.
- Cross-team coordination is decided by the teams. Prefer decentralized and informal coordination over centralized coordination. Emphasize Just Talk and informal networks via communicate in code, cross-team meetings, component mentors, travelers, scouts, and open spaces.
- Product Backlog Refinement (PBR) is preferably done with multiple teams to increase shared learning and to exploit coordination opportunities.
- There is one product Sprint Review; it is common for all teams. Ensure that suitable stakeholders join to contribute the information needed for effective inspection and adaptation.
- Each Team has their own Sprint Retrospective.
- An Overall Retrospective is held after the Team Retrospectives to discuss cross-team and system-wide issues, and create improvement experiments. This is attended by Product Owner, Scrum Masters, Team representatives, and managers (if any).

LeSS Huge Framework Rules

LeSS Huge applies to products with “8+” teams. Avoid applying LeSS Huge for smaller product groups as it will result in more overhead and local optimizations.

All LeSS rules apply to LeSS Huge, unless otherwise stated. Each Requirement Area acts like the basic LeSS framework.

LeSS Huge Structure

- Customer requirements that are strongly related from a customer perspective are grouped in Requirement Areas.
- Each Team specializes in one Requirement Area. Teams stay in one area for a long time. When there is more value in other areas, teams might change Requirement Area
- Each Requirement Area has one Area Product Owner.
- Each Requirement Area has between “4-8” teams. Avoid violating this range.
- LeSS Huge adoptions, including the structural changes, are done with an evolutionary incremental approach.
- Remember each day: LeSS Huge adoptions take months or years, infinite patience, and sense of humor.

LeSS Huge Product

- One (overall) Product Owner is responsible for product-wide prioritization and deciding which teams work in which Area. He works closely with Area Product Owners.
- Area Product Owners act as Product Owners towards their teams.
- There is one Product Backlog; every item in it belongs to exactly one Requirement Area.
- There is one Area Product Backlog per Requirement Area. This backlog is conceptually a more granular view onto the one Product Backlog.

LeSS Huge Sprint

- There is one product-level Sprint, not a different Sprint for each Requirement Area. It ends in one integrated whole product.
- The Product Owner and Area Product Owners synchronize frequently. Before Sprint Planning they ensure the Teams work on the most valuable items. After the Sprint Review, they further enable product-level adaptations.