

Multi Team Scrum (Expansion of the SGEP)

Ralph Jocham

John Coleman

2026-01-18T09:00:00Z

Collected Resources for Scrum Guide Expansion Pack

This document is a collection of independent works. Each section retains its original license or copyright status, as indicated. Please refer to each section for specific usage rights and requirements.

License: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

© 2025 Ralph Jocham, John Coleman, and Jeff Sutherland.

Disclaimer: No warranties are given. Use at your own risk.

This section is offered under the Attribution-ShareAlike 4.0 International license of Creative Commons.

By using this Scrum Guide Expansion Pack, you agree to the terms of the CC BY-SA 4.0 license.

If a Scrum Team becomes too large, it should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same Product. Multiple Scrum Teams on the same Product should share the same Product Goal, Product Backlog, Product Owner, baseline Definition of Outcome Done, and baseline Definition of Output Done.

Be careful with blind assumptions that more value is produced from more Scrum Teams. Scale only when the benefits clearly outweigh the additional overhead. Before you scale, the single Scrum Team setup has to be able to reliably produce an Increment every Sprint. But if you must scale, use an approach that is coherent with this document. Often, fewer teams result in more outcomes.

In a multi-Scrum-Team context, Scrum Teams may reduce inter-Scrum-Team dependencies by becoming more cross-functional through collaboration, cross-pollination, transfer of learning, and intentional interactions. The specific skills needed are often broad and will vary with the domain of work. In a multi-Scrum-Team setting, purposeful and intentional interactions and professionalism (including continuous integration) become even more important.

In a one Product Owner and one Scrum Team setup, the Product Owner could be a Product manager, marketing director, technology director, etc. In a multi-Scrum-Team setup for a Product, the ideal is still only one Product Owner, who should be a leader for the Product. To allow the Product Owner to handle multiple Scrum Teams on the same

Product, the Product Owner often becomes more strategic and delegates problems to solve and opportunities to the Product Developers, including, for example, aspects of Product design or Product management.

The Product Backlog is a tool for increasing transparency.

Generally, the fewer Product Backlogs per Product, implicit (like a filter of a Product Backlog) or explicit:

- The fewer the silos in the Product and the greater the transparency across the entire Product;
- The more transparent the overall progress tracking across the entire Product;
- The better the big-picture value clarity across the entire Product;
- The more likely a Scrum Team would know it's working on low-value items from a Product perspective;
- The more likely one is to observe improvement in the attainment of value; and,
- The more strategic the Product Owner becomes by delegating cross-Product work to the Product Developers.

Fewer Product Backlogs per Product are better for adaptiveness [1], but without empowered ownership, a coherent span of control, or direct contact with relevant Stakeholders, gaps will arise. Scrum fosters a climate for happenstance and multi-learning as various people and Scrum Teams collaborate, discoveries and insights can be shared and leveraged. This is unlikely to happen in an environment where each component has a Product Backlog in isolation.

Happenstance in the context of 'The New New Product Development Game' [2] means that sometimes useful ideas or solutions come about by chance, not by careful planning. When Scrum Teams work closely together and share information, they might discover new approaches or answers simply because they are open to unexpected events or accidental findings.

Multi-learning means that team members learn in many different ways at the same time. They pick up new skills and knowledge not only in their own area but also in other areas, and they learn as individuals, as a group, and as part of the whole company. This helps the team become more flexible and able to solve a wide range of problems quickly, because everyone is learning from each other and from their experiences as they work together.

Finding the right balance is a dilemma. There are always trade-offs to consider. However, a good heuristic is: The fewer Product Backlogs, implicit or explicit, the better, enabled through multi-learning and the organizational transfer of learning across Scrum Teams, departments, and Products.

Organizational transfer of learning, as described in 'The New New Product Development Game' [2], is the process by which knowledge and insights gained in one new Product development area are regularly shared and applied to subsequent areas or other divisions within the organization.

Organizations are often designed for ease of management over ease of outcomes. Ask

yourself how many Scrum Teams a problem or opportunity touches to deliver value; generally, the lower that number, the better.

Free teams from command and control. Err on the side of aligned autonomy. Foster purposeful, intentional interactions within and between self-managing Scrum Teams [3]. Foster a work climate with minimal but sufficient management processes, scaffolding, and boundaries. Balance and nurture Stakeholder expectations and limits. Build change agency and continual improvement in a direction, not just delivery, into a cadence.

When in doubt, study the New New Product Development Game [2], embrace the good of what's new in the present, but abandon any notions of an industrial complex [4-9] where only the brave people have the agency to do anything.

Attribution for the Scrum Guide Expansion Pack Collection

This collection was written and compiled by *Ralph Jocham, John Coleman, and Jeff Sutherland*. Each section is individually attributed above and retains its original license. The collection as a whole is for informational purposes; please respect the license terms of each section.

References

- [1] LeSS (n.d.) 'Why LeSS? Achieving adaptiveness'. Available at: <https://less.works/less/framework/why-less> (Accessed: 17 May 2025).
- [2] Takeuchi, H. and Nonaka, I. (2014) The new new product development game, Harvard Business Review. At: <https://hbr.org/1986/01/the-new-new-product-development-game> (Accessed: 21 January 2024).
- [3] LeSS.works (n.d.) Self-managing teams. Available at: <https://less.works/less/management/self-managing-teams> (Accessed: 17 May 2025).
- [4] Cynefin.io, V. (2022) Cynefin wiki, Cynefin.io. Cynefin.io. At: <https://cynefin.io/> (Accessed: April 4, 2023).
- [5] Rancati, A. and Snowden, D. (2021) Managing complexity (and chaos) in a crisis - a field guide for decision makers inspired by the Cynefin framework. Luxembourg, Belgium: Publications Office of the European Union.
- [6] Snowden, D. et al. (2022) Cynefin® weaving sense-making into the fabric of our world. 2nd edn. Edited by R. Greenberg and B. Bertsch. Singapore, Singapore: Cognitive Edge - The Cynefin Co.
- [7] Snowden, D. (2023) Cynefin St David's 2023 1 of 2, Cynefin Co. <https://thecynefin.co/cynefin-st-davids-2023-1-of-2/> (Accessed: April 20, 2023).
- [8] Snowden, D. (2023) Managing for emergence through abduction, The Cynefin Co. At: <https://thecynefin.co/managing-for-emergence/> (Accessed: June 24, 2023).
- [9] Snowden, D. and Smith, N. (2023) Leadership discussion: Dave and Natalie - the Cynefin co, YouTube. At: <https://youtu.be/WcPZ8ybDF0w> (Accessed: April 7, 2023).