Scrum

Maren Heibges, Katharina Jungnickel, and Markus A. Feufel

Definition

Scrum is a project management method. Transdisciplinary group learning and collaborative research, especially communication and coordination, can be fostered by adapting the method. The term Scrum originally derives from the rugby ball game, where it refers to a tight huddle when the game restarts (Cervone 2011, 19), and has entered into the (project-)management literature through an article by economists Takeuchi and Nonaka (1986), where they describe new, more flexible and incremental approaches to product development they observed in Japan and the USA in the mid-1980s.

In project management, Scrum's main features are formalized forms of communication and coordination, which require relatively little documentation, a strict meeting rhythm, and particular management roles and artifacts (e.g. a progress board) to manage work tasks (Schwaber and Sutherland 2020). Within this rather rigid framework, Scrum promises flexible, non-hierarchical, and self-organizing work processes thanks to built-in, self-correcting feedback loops for small teams of five to ten people (e.g. Pope-Ruark 2012).

Specifically, project management with Scrum relies on three defining features (Preußig 2015; Schwaber and Sutherland 2020; Shalloway et al. 2010). The first defining feature relates to cyclical workflows, that is, meetings that are repeated after fixed time intervals, which are called *sprints* in Scrum terminology. In contrast to classical project management, Scrum specifies most planning details "on the way". Rather than planning years ahead, which is time-consuming and makes it hard to adapt to new circumstances (Pope-Ruark 2017, 10–13), Scrum needs just a broadly defined general goal at the outset, which is spelled out in detail and revised if needed as the project evolves. This self-corrective function is at the root of the cyclical workflow described above, ultimately enabling adaptive work processes, often referred to as *agile* project management (e.g. Wintersteiger 2015, 20).

Depending on the timeframe of a specific project, work cycles or *Sprints* typically last between one to six weeks and consist of the following meeting formats, which, depending on work cycle length and project size, can each be adapted in length:

- *Planning meeting or Sprint planning:* In this meeting, the project team specifies the tasks necessary for completing the work package that has been prioritized for the upcoming work cycle.
- *Review meeting or Sprint review:* During this format the outcomes of the previous work cycle are reviewed by the team and relevant stakeholders. This meeting aims to evaluate project progress.
- *Introspection meeting or Sprint retrospective:* This is an introspective format, where the team evaluates the past work cycle. Main outcomes of this meeting are decisions about what should be changed in the following work cycle.
- *Update meeting or daily Scrum:* Unlike the other meetings, which are scheduled once every work cycle, this meeting takes place every day. It is only about 15 minutes long and conducted as a stand-up meeting. Each team member identifies what has been completed since the last update, what task they will take on next, and whether there are any obstacles to completion. It serves to synchronize the team members and identify any obstacles.

Once meeting timetables have been specified by the team, they cannot be changed for the ongoing work cycle. This practice points to one essential aspect of the Scrum framework: extremely strict time-keeping referred to as "time-boxing" (e.g. Fowler 2019, 75–76) applied to all Scrum meetings. For instance, if a team runs into problems midway into a work cycle, then the planned timeframe of the work cycle cannot be extended. Instead, the following work cycle will be planned with the past delays in mind. In other words, milestones are more flexible in Scrum and timeframes are more rigid than in classical project management.

The second key component of Scrum is the use of artifacts, such as an (analog or digital) whiteboard to track project progress. This whiteboard contains all tasks identified for a work cycle during the planning meeting (e.g. on sticky notes) and visualizes task progress by the position of these notes in one of three predefined columns, labeled "to do", "in progress", and "done".

The only more conventional documentation format used in Scrum is a project logbook, called *Product Backlog*, which is a list of all work packages to be completed to achieve the overall project goal. Importantly, these work packages are not yet planned out in detail but only roughly defined and prioritized. Over time, as the project progresses, the work packages are broken down into individual tasks during the planning meetings. In the logbook, the goals of individual work packages are usually described by one or two simple sentences, called a *User Story*, to facilitate a shared understanding between team members. A similar communication technique is used when team members define and agree on a short and easy-to-understand *Definition of Done* for each individual task that has been identified as part of a work package.

The third defining feature of Scrum is, in contrast to classical approaches, that it designates two types of project manager roles (Schwaber and Sutherland 2020, 5–7). There is a project facilitator, referred to as the *Scrum Master,* and a project planner, who is called the *Product Owner* in Scrum terminology. The project facilitator runs all meetings, is responsible for solving problems that hinder the workflow, and makes sure that the team complies with the Scrum formats. This role necessitates dominant facilitation techniques to ensure that the Scrum framework stays intact, and it entails significant amounts of troubleshooting to ensure the work processes run smoothly. The project planner, in comparison, focuses on maintaining and refining the project goals and priorities in the logbook by planning and revising necessary next steps *beyond* the current work cycle. The person assuming this role is also in charge of communicating about the project with external stakeholders, such as academic partners or clients.

Background

By the mid-1990s, Schwaber and Sutherland introduced Scrum as a project management framework for software development, which has been updated several times since (Schwaber and Sutherland 2020). From the outset, the design of Scrum has been driven by the desire to operationalize principles of agile management, such as flexibility, transparency, and incremental improvement (Beck et al. 2001). Scrum's ability to foster self-correcting workflows (in other words, its potential to foster agility) is one of its most acclaimed strengths (Wintersteiger 2015). This flexibility is contrasted with more classical approaches to project management, most notably the *Waterfall Model* (e.g. Thesing et al. 2021). Waterfall planning leads to a detailed and fixed project schedule, for instance through highly specified project milestones, which are identified in the planning phase even before the start of the project.

Today, most software is developed within an agile framework (e.g. Krawczyk-Bryłka and Krawczyk 2019), which is based on different forms of implementation, such as Kanban, Extreme Project Management, and Scrum as the most popular method (e.g. Cervone 2011, 19). Given the success of Scrum in the domain of software and product development, it is now being promoted as a general method for project management in many areas of project work, also for higher education and research (Pope-Ruark 2017). However, the literature suggests that Scrum is thus far rarely used in academia (e.g. Fernandes et al. 2021, 4). Most papers discussing applications of Scrum in higher education are low-profile proceedings from international computer science or engineering conferences (e.g. Hicks and Foster 2010; Linos et al. 2020; Ochoa et al. 2021; Persson et al. 2011), with only a few exceptions stemming from educational science and discussing academic applications of Scrum in a non-technical academic setting (e.g. Fernandes et al. 2021; Pope-Ruark 2012, 2017).

Despite the dearth of research, and although Scrum was originally developed to coordinate project management for monodisciplinary teams of software developers (e.g. Schwaber and Sutherland 2020, 5), the literature and the authors' facilitation experience with Scrum suggests that the method can be adapted to create and maintain collaborative accountability (i.e. a shared understanding of goals, processes, and data) in transdisciplinary learning and research contexts. In particular, Scrum can synchronize academic collaboration – acutely so for transdisciplinary teams that are marked by diverse bodies of knowledge and skill sets – through its formalized communication techniques and a focus on creating and maintaining shared (visual) representations of collaborative workflows on progress boards. In other words, Scrum can guide the communication and organization necessitated by transdisciplinary group learning and research.

Debate and criticism

Scrum is mostly criticized for potential shortcomings in oversight, because no detailed milestone roadmap or waterfall project plan is followed (e.g. Cervone 2011, 22; for a general debate see Serrador and Pinto 2015). This is particularly problematic for project management in academia as most funding agencies require standard waterfall plans and reporting. Some practitioners of Scrum also criticize the framework for introducing a certain "breathlessness" into project work, referring to the repetitive work cycles and the metaphorical and literal emphasis on sprinting through fixed time intervals. For academic projects, this may not provide enough time to think and focus on in-depth analyses. Furthermore, Scrum language is rather inaccessible to those unfamiliar with the framework, hampering the creation of a common language and common ground, which already is a challenge for inter- and transdisciplinary team work in academia. Finally, the Scrum philosophy or mindset – which is emphasized as one of its most central parts (e.g. Sloan 2015) – might be perceived as too dogmatic by academic teams and may lead to resistance rather than effective work processes. There are also voices, however, which underline the (politically) progressive potential of the Scrum mindset to foster non-hierarchical work environments and a communicative work and feedback culture (e.g. Pope-Ruark 2017, 15–22).

Project facilitation via Scrum is increasingly in demand for its potential to coordinate distributed teams (e.g. Fowler 2019; Krawczyk-Bryłka 2017) – increasingly so in the wake of the worldwide trend towards remote work during and after the Covid-19 pandemic (Henke et al. 2022). Generally, Scrum seems to function just as well, or even more effectively, for distributed and hybrid teams (Sutherland et al. 2007). However, more technology-prone or introvert team members may fare better in a digital (Scrum) environment than technology-averse or extrovert team members, who may prefer analog interactions (e.g. Grelle and Popp 2021). Similarly, team members who interact in highly diverging intensities with digital progress boards require increased coordinative efforts (Hidalgo 2019). Finally, digital facilitation, for Scrum or other formats, requires more planning and active chairing than face-to-face facilitation to avoid what has become known as "Zoom fatigue" (e.g. Nesher Shoshan and Wehrt 2022).

On the positive side, Scrum has the potential to foster team work and systemize the management of group learning, research, and administration (e.g. Hidalgo 2019; Pope-Ruark 2017). For the university classroom, there is evidence that the transparent and ritualized communication and planning and the clear role allocation in Scrum may foster a collaborative learning environment and reduce student anxiety related to group dynamics and the rejection of group projects (Allan 2016; Fernandes et al. 2021; Pope-Ruark et al. 2011).

Especially in transdisciplinary academic contexts, where students, external stakeholders, and researchers do not share the same knowledge and background, formalized communication techniques may help to keep misunderstandings at bay and to synchronize goals and achievement strategies. For instance, by collectively negotiating a *Definition of Done* in a transdisciplinary group, possibly diverging expectations about the goals ("what do we need to do"), the methodology ("how do we get there?"), and the medium of the final product ("how do we (re-) present our results") can be identified – and clarified if needed – as a basis for effective team work.

Finally, the lack of agility in higher education tends to be a general obstacle to introducing Scrum to academia: Typically, higher education is not particularly flexible when it comes to changing research goals or curricula to meet new requirements. For example, in funding applications, project goals usually need to be mapped out using Gantt charts and underpinned with concrete project milestones, even if the project is to be managed with Scrum. And both the university classroom and the academic research group are typically marked by steep hierarchies in contrast to the flat hierarchies envisioned for effective cooperation and coordination in Scrum teams (Schwaber and Sutherland 2020).

Current forms of implementation in higher education

There are currently two forms of implementation of Scrum in higher education: (1) to help manage transdisciplinary research (*Scrience*) and (2) to facilitate collaborative and transdisciplinary learning (*EduScrum*).

To manage transdisciplinary research groups, Scrum provides a systematic approach to collaboration and communication, which is used to some extent in IT or engineering departments and rarely in domains outside of this context (but see Hidalgo's 2019 case study on using Scrum in a UK policy research center). To provide insights into the more general forms of implementing Scrum to manage research, the authors of this chapter therefore share their own practical experience with devising, introducing, and maintaining Scrum to manage transdisciplinary research projects (see Speiser et al. 2023). Given that the developers of Scrum, Schwaber and Sutherland (2020, 13), are adamant that their framework should *not* be called Scrum if it was changed in any way, the authors refer to their implementation as *Scrum for Science* or *Scrience*. However, the Scrum framework can and should be maintained when implementing *Scrience*.

To account for the fact that *Scrience* is not only about practice and products (as the original Scrum) but also about knowledge, education, and discourse, *Scrience* requires an additional "science meeting" to formalize discussions of scientific theories, findings (e.g. newly collected data), and thesis work. The science meeting should be scheduled in the middle of every work cycle. To account for the busy work schedules of academics, which often involve more than one project at a time, work cycles should be extended (e.g. to about three to four weeks) to create the right meeting density that fits into typical faculty or student schedules (see also Hidalgo 2019, 17). For similar reasons, the update meetings should be scheduled weekly rather than daily (see also Baham 2019, 142; Ochoa et al. 2021, 4).

For *Scrience* to succeed, a second critical feature is that members of academic teams possess, on the one hand, unique and diverse expertise instead of similar skills as originally envisaged in the Scrum framework (Hidalgo 2019, 18–19) and may assume multiple roles on the other. To account for the uneven distribution of expertise, it is recommended to treat each task *as if* it was for a generalist team to ensure transparency, team synchronization, and consensus, even if certain tasks realistically can only be completed by an individual team member. To avoid role confusion, the role of project facilitator (Scrum Master) should be assigned to a senior team member with authority in the team rather than to a student assistant, and with sufficient resources to regularly chair the meetings (see also Baham 2019, 150). For similar reasons, the Principial Investigator (that is, the person in charge of the research project) should take on the role of project planner (Project Owner), who coordinates and prioritizes work packages with the overall project goal in mind. If the Principal Investigator does not have the time to be actively involved in project planning, this role can also be performed by a team member capable of the necessary scientific oversight. However, the Principal Investigator must then accept that this practice reduces his or her role to a project stakeholder, who is only occasionally consulted for input by the chosen project planner (see also Fowler 2019, 74).

Finally, given that *Scrience* is a new form of project management that most team members will be unfamiliar with, every project needs an extended introductory phase during which all team members are familiarized with the main concepts and are given the opportunity to practice the steps involved and make mistakes (also see Pope-Ruark 2012, 165–67). To facilitate transdisciplinary communication, it is recommended to use accessible language instead of potentially inaccessible Scrum terminology for the different components of *Scrience*, such as speaking of 'work cycles' instead of 'Sprints' – as already demonstrated throughout this chapter. During the introductory phase, the team members should also decide on a meeting rhythm and work cycle length and commit to keeping this schedule for at least three to six months before revising it if necessary. As for the original Scrum, Scrience works best for groups of four to ten members. Larger teams may be divided into separate smaller groups with different thematic focus if needed (e.g. on technology development versus study design; also see Schwaber and Sutherland 2020, 5).

To manage collaborative and transdisciplinary learning, Scrum's most common form of implementation is in the context of applied engineering and computer science courses around the world, where students work on development projects and learn Scrum as a project management tool on the side (Baham 2019). This practice is sometimes referred to as *EduScrum* (e.g. Fernandes et al. 2021; Neumann and Baumann 2021). EduScrum has also been used successfully outside of engineering curricula and with the broader goal of enabling project-based, collaborative, and transdisciplinary student learning in the academic classroom. Pope-Ruark (2012, 2017) has shown, for instance, that Scrum can be used to manage student projects in the humanities just as much as in engineering or programming courses.

The focus of EduScrum mostly lies on equipping students with Scrum as a skill, which they will need later in industry and to manage practical course work effectively (e.g. Kudikyala and Dulhare 2020). In this context, the lecturer often assumes the role of the project facilitator (e.g. Persson et al. 2011, 63; Pope-Ruark 2017, 171–74), that is the role of the Scrum Master, whereas project planning is done by a student team member or by an external stakeholder from industry. EduScrum may thus be an effective way to satisfy the increasing need for project management tools in the classroom arising from a general didactic trend towards group learning in cooperative education, design thinking, storytelling, student-organized teaching, and transdisciplinary learning in general.

In summary, Scrience and EduScrum effectively adapt the original Scrum to fit individual educational and scholarly needs. Although both frameworks are helpful tools to promote transdisciplinary research and learning by providing a clear and ready-to-implement collaboration and communication framework for interdisciplinary teams, small deviations from the described approaches matter and may alter their effectiveness. Thus, in the future both Scrience and EduScrum might benefit from further evidence-based consolidation in the form of published quality standards or – using the language of agile management (Schwaber and Sutherland 2020; Takeuchi and Nonaka 1986) – more definite "rules of the game".

References

Allan, Elizabeth. 2016. "I hate group work!": Addressing students' concerns about small-group learning. *InSight: A Journal of Scholarly Teaching* 11: 81–89.

Baham, Corey. 2019. Implementing Scrum wholesale in the classroom. *Journal of Information Systems Education* 30 (3): 141–59.

Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas. 2001. *Agile manifesto*. Available from https://agilemanifesto.org.

Cervone, H. Frank. 2011. Understanding agile project management methods using Scrum. *OCLC Systems & Services* 27 (1): 18–22.

Fernandes, Sandra, José Dinis-Carvalho, and Ana Teresa Ferreira-Oliveira. 2021. Improving the performance of student teams in project-based learning with Scrum. *Education Sciences* 11 (8): 2–16.

Fowler, Frederik M. 2019. *Navigating hybrid Scrum environments*. 2018. Berkeley: Apress.

Grelle, Darrin, and Eric Popp. 2021. Considering the interaction of individual differences and remote work contexts. *Industrial and Organizational Psychology* 14 (1–2): 244–47.

Henke, Jonn B., Samantha K. Jones, and Thomas A. O'Neill. 2022. Skills and abilities to thrive in remote work: What have we learned? *Frontiers in Psychology* 13: 893895.

Hicks, Michael, and Jeffrey S. Foster. 2010. SCORE: Agile research group management. *Communications of the ACM* 53 (10): 30–31.

Hidalgo, Enric Senabre. 2019. Adapting the Scrum framework for agile project management in science. *Heliyon* 5 (3): e01447.

Kudikyala, Udai Kumar, and Uma N. Dulhare. 2015. Using Scrum and Wikis to manage student major projects. *International Conference on MOOCs, Innovation and Technology in Education*, eds. Deepak Garg, Russ Meier, and Rob Reilly, 15–20. Available from https://ieeexplore.ieee.org/document/7375279.

Krawczyk-Bryłka, Beata. 2017. Comparative study of traditional and virtual teams. *TASK Quarterly* 21 (3): 233–45.

Krawczyk-Bryłka, Beata, and Henryk Krawczyk. 2019. The preferable ways of decision-making in IT teams. *Decision Making in Manufacturing and Services* 13 (1–2): 37–51.

Linos, Panagiotis K., Ryan Rybarczyk, and Nathan Partenheimer. 2020. Involving IT professionals in Scrum student teams. *Frontiers in Education Conference*. Available from https://doi.org/10.1109/FIE44824.2020.9274190.

Nesher Shoshan, Hadar, and Wilken Wehrt. 2022. Understanding "Zoom Fatigue": A mixed-method approach. *Applied Psychology* 71 (3): 827–52.

Neumann, Michael, and Lars Baumann. 2021. Agile methods in higher education: Adapting and using EduScrum with real world projects. *IEE Frontiers in Education Conference*. Available from https://doi.org/10.1109/FIE49875.2021.9637344.

Ochoa, Omar, Massood Towhidneiad, Timothy Wilson, James Pembridze, Erin Bowen, and Carlos Castro. 2021. *Adopting agility in academia through pilot projects.* Available from https://ieeexplore.ieee.org/document/9637458

Persson, Mia, Ivan Kruzela, Kristina Allder, Per Johansson, and Olof Johansson. 2011. On the use of Scrum in project driven higher education. *Proceedings of the 2011 International Conference on Frontiers in Education Computer Science & Computer Engineering*, ed. IEEE Computer Society, 59–64. Waikoloa, HI: IEEE. Available from http://worldcomp-proceedings.com/proc/p2011/FEC3416.pdf.

Pope-Ruark, Rebecca. 2012. We Scrum every day. *College Teaching* 60 (4): 164–69.

Pope-Ruark, Rebecca. 2017. *Agile faculty*. Chicago: University of Chicago Press.

Pope-Ruark, Rebecca, Michelle Eichel, Sarah Talbott, and Kasey Thornton. 2011. Let's Scrum: How Scrum methodology encourages students to view themselves as collaborators. *Teaching and Learning Together in Higher Education* 3 (3): 1–16.

Preußig, Jörg. 2015. *Agiles Projektmanagement*. Freiburg: Haufe.

Schwaber, Ken, and Jeff Sutherland. 2020. The Scrum guide. Available from https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.

Serrador, Pedro, and Jeffrey K. Pinto. 2015. Does agile work? A quantitative analysis of agile project success. *International Journal of Project Management* 33 (5): 1040–51.

Shalloway, Alan, Guy Beaver, and James Trott. 2010. *Lean-agile software development*. Upper Saddle River, NJ: Addison-Wesley.

Sloan, Dan. 2015. *Is the "growth mindset" an agile mindset?* Available from https://www.scrum.org/resources/blog/growth-mindset-agile-mindset.

Sutherland, Jeff, Anton Viktorov, Jack Blount, and Nikolai Puntikov. 2007. Distributed Scrum: Agile project management with outsourced development teams. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, ed. IEEE Computer Society, 274a. Waikoloa, HI: IEEE.

Speiser, Dorothee, Maren Heibges, Laura Besch, Caren Hilger, Marie Keinert, Katharina Klein, Christine Olbrich, Julia Kußmaul, Caroline Neeb, Gudrun Rauwolf, Christine Schmid, Stephanie Stegen, Viola Westfal, Isabell Witzel, Benedikt Zang, Friederike Kendel, and Markus A. Feufel. 2023. iKNOW – A paradigmatic approach to support personalized counseling with digital health. *JMIR Formative Research 2023;7:e41179*. Available from https://www.doi.org/10.2196/41179.

Takeuchi, Hirotaka, and Ikujiro Nonaka. 1986. The new product development game. *Harvard Business Review* 64 (1): 137–46.

Thesing, Theo, Carsten Feldmann, and Martin Burchardt. 2021. Agile versus waterfall project management. *Procedia Computer Science* 181: 746–56.

Wintersteiger, Andreas. 2015. *Scrum*. Frankfurt am Main: entwickler.press.