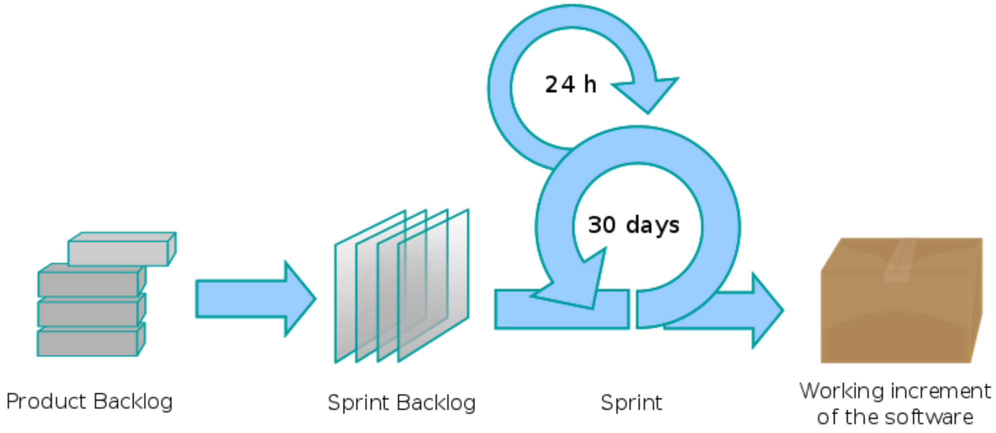
www.datascience-pm.com /scrum/

Scrum – Data Science Project Management

11-14 minutes

Scrum Overview

Co-founded by Jeff Sutherland and Ken Schwaber in the 1990s, Scrum has become the most commonly used agile approach with over 12 million practitioners (scrum.org). Although heavily adopted in software, Scrum is also used across a wide variety of companies. National Public Radio uses it to create new programming, John Deere for new machinery development, Saab for fighter jets, and C.H. Robinson applies Scrum for human resources (Rigby, Sutherland, & Takeuchi, 2016).



Scrum Process Diagram (Wikimedia Commons, 2008)

The Scrum Guide is the definitive guide to Scrum. It recommends teams of 3 to 9 development members and defines three roles (Sutherland & Schwaber, 2017):

- **Product Owner:** sets product vision and defines potential product increments (also known as *user stories* or *features*). The list of such increments is called the *product backlog*
- **Scrum Master:** facilitates the Scrum process as a servant leader
- **Development Team:** professionals who deliver product increments; in the context of data science teams — data scientists, data engineers, data analysts, systems analysts, and software engineers

Scrum divides the larger project into a series of mini-projects, each of a consistent and fixed length ranging from one week to one month. Each mini-project cycle, called a *sprint*, kicks off at a meeting called *sprint planning* where the product owner defines and explains the top feature priorities. The development team forecasts what increments they can deliver by the end of the sprint and then makes a sprint plan to develop these increments. During the sprint, they coordinate closely and develop daily plans at *daily standups*. At the end of the sprint, the team demonstrates the increments to stakeholders and solicit feedback during *sprint review.* These increments should be *potentially releasable* and meet the pre-defined *definition of done.* To close a sprint, the team inspects itself and plans for how it can improve in the next sprint during the *sprint retrospective* (Sutherland & Schwaber, 2017).

Scrum is the de facto agile project management framework, so much so *agile* and S*crum* are synonymous in the minds of several professionals. Although several interviewees expressed that they are using Scrum, none were following the full framework. Likewise, software engineering teams who “use scrum”, tend to deviate slightly or even extensively from its processes and principles. Scrum purists may frown upon deviations, but others view these as appropriate to cater to individual team needs.

Scrum is often combined with other project management approaches. Most notably, development teams often coordinate development tasks during each sprint using a Kanban board.

Does Scrum Work for Data Science?

When asked if Scrum is appropriate for data science, one interviewee quipped that: "I've yet to find anything better" (Jennings, 2017). This response seems to sum up the common perceptions that Scrum is not ideal for data science, another approach would probably be better but has yet to be discovered, but that Scrum generally works better than known alternatives.

Benefits

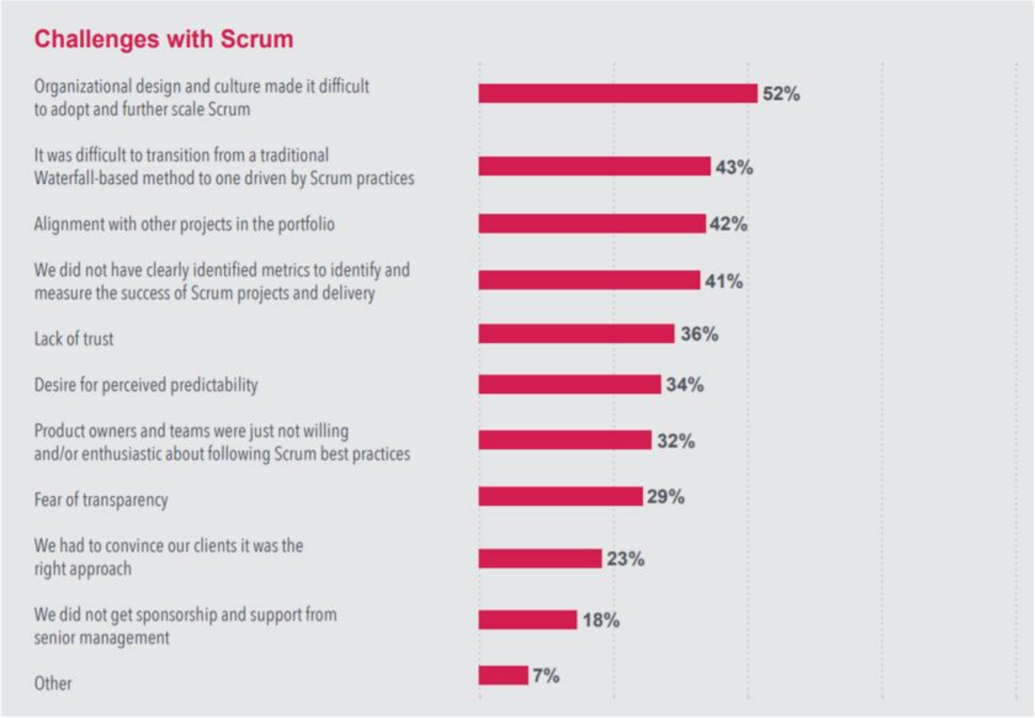
- **Customer Focus:** Scrum focuses on delivering customer value. It frames development work through the lens of the customer and encourages stakeholders to provide constant feedback through practices such as sprint reviews. In a survey, executives rated "Delivering business value to the customer" as the most valued aspect of Scrum projects (Scrum Alliance, 2017).
- **Regularity:** Its rigid time boundaries allows teams to get used to flow through a regular work cadence – something that caters to basic human psychology (Sutherland, 2014).
- **Autonomy:** By providing teams with broad autonomy to self-govern, they are happier, more productive, and more engaged (Pink, 2009) (Sutherland, 2014).
- **Improvement through Inspection:** Teams are called to constantly inspect themselves, especially during retrospectives at the end of each sprint which can help them to accelerate their performance through each development cycle by learning and adapting from the previous cycles (Sutherland & Schwaber, 2017).
- **Empirical Evidence:** Like data science, Scrum is founded on the principle of execution based on what is known. "Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk"(Sutherland & Schwaber, 2017).
- **Well-known:** As the most common agile approach with over 12 million practitioners (Scrum Alliance, 2017), team members might already have familiarity and experience with Scrum which eases Scrum adoption.
- **Accountability:** Individuals feel accountable to their team members. This is reinforced by the transparent nature of Scrum and practices such as the daily standup and sprint reviews where team members are constantly demonstrating their work (Sutherland & Schwaber, 2017).
- **Sense of Urgency:** Because a deadline is always looming, teams may be motivated to deliver. Procrastination will become painfully obvious.

Weaknesses and Challenges

Scrum is controversial, largely due to its rigid time boxing and organizational/cultural conflicts.

- **Time Boxing Challenges:** Although it provides numerous benefits, the time boxing feature of Scrum is controversial, especially for data science teams. The time required to implement a solution for most data science problems is ambiguous. Teams that use Scrum typically go through estimation exercises, often in so-called *story points*, to assign a rough work size estimate for a feature. Asking data scientists to provide such estimates may be an awkward exercise as the estimates depend on unknowns such as data quantity, data quality, or the untested ability to differentiate signal from noise. Moreover, data scientists might be uncomfortable committing that they will complete certain features by the end of a sprint or become frustrated if they are unable to hit the commitments they forecasted (Clerkin, 2017). When pressed to hit the deadline, teams might cut corners – poor quality from insufficient documentation, incomplete testing, and unnecessary technical debt may accrue as a result.
- **Cultural Challenges:** From a corporate culture perspective, Scrum is unsettling to many across all levels of the organization (Rigby, Sutherland, & Takeuchi, 2016). It tends to upend the traditional hierarchical reporting structure of organizations. Development team members who are used to having their work directed to them might become uncomfortable with self-directed work responsibilities. Previously siloed data scientists might also indirectly oppose the emphasis of teamwork. Management, who traditionally hold much authority over teams, may oppose yielding responsibilities that allow teams to self-organize. Executives might also demand well-defined, long-term project timelines which are not feasible in Scrum. In general, outsiders to Scrum teams work off different planning cycles. Collectively, these issues present numerous cultural adoption issues to Scrum. Such, "organizational design and culture" issues are the most commonly stated challenge (52% of respondents) in implementing Scrum (Scrum Alliance, 2017).

- **“Potentially Releasable Increment” Challenges:** Scrum prescribes teams to deliver “potentially releasable Increment of ‘Done’ product” at the end of each sprint (Sutherland & Schwaber, 2017). However, Sutherland reports that “80% of the hundreds of Silicon Valley teams I polled could not do this” (Sutherland, 2015). This challenge is particularly daunting for data science teams and might not even be necessary. Much of the data science process, especially in the early exploratory phases, may not be intended for release outside the data science team. Moreover, testing requirements may extend beyond what is reasonable in a sprint, especially if a high degree of accuracy is needed (Jennings, 2017).
- **Meeting Overhead:** Scrum’s meetings may require up to four hours per week, plus additional time for story refinement and backlog management (Sutherland & Schwaber, 2017). These meetings might be viewed as overhead that should be avoided (Beverly, 2017).
- **Difficult to Master:** The Scrum Guide (Sutherland & Schwaber, 2017) starts out stating that Scrum is “difficult to master” and survey respondents in the State of Scrum Report (Scrum Alliance, 2017) would agree. For example, student teams working on a data science project in a controlled experiment who used Scrum performed the worst, largely because of their inability to understand the methodology and to set up clear sprints (Saltz, Shamshurin, & Crowston, 2017).



Results from a 2016 survey in State of Scrum Report

Recommendations

Data science Scrum teams should pro-actively acknowledge these challenges and devise plans to accommodate Scrum for their needs. Possible solutions include:

- **Provide Training:** Organizations that implement Scrum should provide extensive training to team members and at least some training to the broader organization (Rigby, Sutherland, & Takeuchi, 2016).
- **Prioritize “Spikes”:** To allow for research and discovery, teams could create *spikes* which are items in the backlog that provide research time for intense study on specified topics. These spikes sit along-side product increment ideas in the backlog. When brought into the sprint, they are considered “done” when the specified research objectives are met, or the time limit expires (Mezick, 2017).
- **Divide Work “Smaller” and Conquer:** Data scientists frequently complain that their work is too ambiguous to estimate how much effort is required before completion. A possible solution is to divide the work increments into smaller pieces that are definable and estimate-able (Mezick, 2017).
- **Shorten Sprints:** Similarly, Daniel Mezick states that “the closer something flirts to chaos, the way to figure out what to do is through frequent inspection.” He recommends dividing sprint cycles into shorter time periods to force these more frequent inspections (Mezick, 2017).

- **Occasionally Relax "Definition of Done":** To avoid full-blown testing for exploratory work and proofs of concept where speed to delivery may be more important than being "done", teams could agree to relax the *definition of done* for certain stories. However, teams should ensure that they don't fall down a slippery slope of accepting lower quality output for core deliverables.
- **Renegotiate Work during the Sprint:** Contrary to popular misconception, the sprint plan is not locked in stone at sprint planning. Rather, "Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned" if it does not "endanger the Sprint Goal" (Sutherland & Schwaber, 2017).
- **Build an "Architectural Runway":** Teams that need a new larger-scale architecture may find careful upfront planning more effective than allowing architecture to develop through emergent design. Diverging from the customer-centric focus of Scrum, data science teams may take a concept from *Scaled Agile Framework (SAFe)* and dedicate some initial sprints to develop an architecture for themselves (Scaled Agile, Inc, 2017).
- **Do the Least Understood Work First:** To reduce risk, a data science team could focus early development cycles on exploratory work and proofs of concept that allow them to get familiar with the data. Akred recommends his teams to "Frontload the parts you don't understand and then as we get confident in our ability we start building the things around it that actually turn that thing into a usable system" (Akred, 2015). If the team is unable to prove feasibility within a reasonable number of cycles, it can re-focus onto other work and avoid unnecessary losses.
- **Integrate with CRISP-DM:** To address the data science process, CRISP-DM can be integrated with Scrum (discussed in emerging approaches) to manage data science projects.

<Previous: Agile Approaches | Next: Kanban >

References