

Agile Foundation: Scrum

David Norton

Scrum provides an agile project management method focusing on the delivery of business value through empirical control and planning techniques. It can be used to complement other agile methods, such as extreme programming and lean software development.

Key Findings

- Scrum is a lightweight agile project management methodology.
- Scrum focuses on the regular delivery of software that provides business value within an empirical control framework.
- Scrum does not provide developer-level best practices and should be paired with an agile method that does.

Recommendations

- Understand scrum's principles, practices and roles.
- Assess and compare your development methods and practices before adopting scrum.
- Gain commitment from senior management and business users to ensure scrum's success. Embed behavioral change.
- Invest in formal scrum and scrummaster training. There is no shortage of scrum training providers.

WHAT YOU NEED TO KNOW

Scrum is an agile development method that focuses on how projects are planned, organized and delivered. It complements other agile development methods, such as extreme programming, which focuses on the engineering disciplines within a context of scrum management. Scrum provides a way to organize teams to make them more productive and produce higher-quality software, while delivering business value. It also helps the team to respond rapidly to changing customer needs because development occurs in cycles of 30 days or less — typically 14 days

Scrum is well placed to be used as an enterprise-class agile development method. Its use continues to expand beyond standard development into legacy and package application, and even off-shore. Scrum practices have been adopted by infrastructure and operation, albeit at a slower pace.

ANALYSIS

Context

Organizations that want to move away from a prescriptive development life cycle must understand the principles and concepts of the alternative agile methods. Development and application managers with a good grasp of agile method will increase their likelihood of success. They will reduce the risk of adopting an agile method with concepts that do not fit with their organizations or product types.

Analysis

Background

Hirotaka Takeuchi and Ikujiro Nonaka first used the term "scrum" to refer to small, high-performing teams in an article titled "The New Product Development Game," which was published in the Harvard Business Review in 1986. They felt that these teams came together in much the same way as a scrum does in the game of rugby. However, the scrum framework did not come into existence until years later. Jeff Sutherland is credited with developing the process in 1993, while he worked at Easel. Ken Schwaber was also working on these principles at the same time, and his publication of "Scrum Development Process" in 1995 first formalized the process.

Approach

Traditional prescriptive software engineering methods are based on the premise that software can be delivered in a predictive manner. Scrum takes an empirical approach and constantly adjusts and fine-tunes the project based on the actual situation, instead of a previously predictive one.

Scrum is based on development cycles of 30 days or less, which are known as sprints. Each sprint is focused on a set of goals and aims to add value by delivering new functions and improvements. At the end of each sprint, the resulting software is released to production, assuming the release makes sense from a business perspective. Alternatively, sprint deliverables may be aggregated, and they will include two or three sprints before they are released to production and to customers.

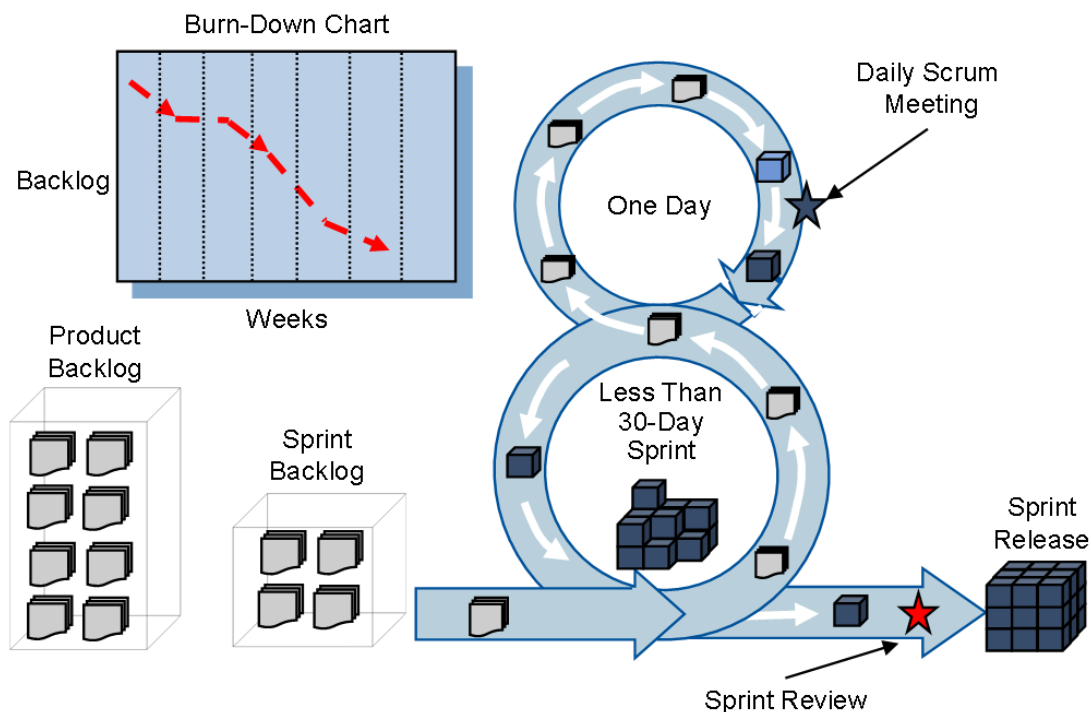
A product owner takes the business requirements and creates a to-do list, which is known as product backlog. The priorities of the tasks are reviewed regularly. The most-important ones are put into a sprint backlog — that is, the tasks the development team must focus on during the next sprint.

The scrum team meets the product owner to conduct a planning meeting before each sprint to review release plans, choose priorities and make any necessary adjustments to what is to be developed. The software is then developed in the sprint phase. During the sprint, a backlog item can undergo any of four activities: development, wrapping, review and adjustment. There is another review meeting at the end of the sprint to assess progress. This usually includes representatives from the business units, who view a demonstration of the software. The outcomes from this meeting act as a basis for planning the next sprint.

This pattern can be repeated as many times as necessary to create the final product, but scrum does not allow the delivery date to change. If the team is behind schedule, then items are deleted from the sprint backlog. If the team is ahead of schedule, then more items are requested.

Figure 1 shows how the scrum life cycle works.

Figure 1. Scrum Life Cycle



Source: Gartner (August 2010)

Roles

There are three main roles within scrum:

Product Owner — This individual represents the customer and ensures that the business needs are met. The product owner is responsible for creating the product backlog and highlighting the highest priorities of the project, which should usually have the highest market value. Other responsibilities include balancing the needs of all stakeholders, defining the features of the product, determining release dates, altering feature requirements and priorities with every sprint, and deciding whether the work of the scrum team is acceptable. The product owner is also in charge of running the sprint planning meetings, but should be available to the scrum team at any time. It should be noted that, despite all these responsibilities, the product owner does not manage the team, because it is self-organizing.

Scrummaster — The scrummaster works closely with the product owner and scrum team in a facilitator role. The aim of this role is to ensure that the team has the right environment in which to function as efficiently and productively as possible. This involves removing any barriers to development, ensuring that lines of communication remain open, and that project information is up to date and available to all interested parties. The scrummaster also protects the team by shielding it from outside management pressure. There may also be a need to resolve any personal problems or conflicts within the team. The scrummaster ensures that the correct processes are followed and is responsible for running daily scrum meetings, as well as the sprint planning and review meetings. This enables the scrummaster to keep track of how much progress is made with the sprint tasks and to update the burn-down chart.

Team — The scrum team consists of the people who will do the development work, such as designers, engineers, architects, testers and programmers. The team is cross-functional and usually consists of five to nine members. Because it is self-organizing, the team selects its own sprint goals and determines what will be produced and how best to achieve this. Ideally, all team members are based in the same room to aid communication and enhance productivity. There are no set roles within the team, so members can swap roles as they see fit.

Artifacts

The most-important artifacts within the scrum process are:

Product Backlog — This contains all the system requirements and lists them in order of priority. Priorities are determined by their value to the customer. The backlog includes all the features that are required and the technical requirements of these features. Contributions to this list can come from within the team or anywhere else in the business, but the product owner has overall responsibility for maintaining this list and ensuring that the highest priorities are met. The product backlog also includes estimates of how long items will take to implement. Each item is defined as a unit of work that is small enough to be completed during a sprint phase.

Release Backlog — This is similar to the product backlog but limits its scope to an actual release of the product being developed. It tends to include more detail about the actual requirements and time estimates provided by the team members.

Sprint Backlog — This is an outcome of the sprint planning meeting. It lists the tasks that must be completed to implement a feature during the upcoming sprint. These tasks are broken down into tasks that can be completed within 16 developer hours. Product backlog items are transferred onto this list once the product owner has chosen them as the highest priorities that need to be focused on during the sprint.

Burn-Down Chart — These are time-based charts that show how much work remains. The amount of work remaining is shown on the Y axis, and time is shown on the X axis. Every scrum project will have two burn-down charts: a *sprint burn-down chart*, which charts daily progress, and a *product burn-down chart*, which shows the overall progress of the development project.

Ceremonies

There are a number of meetings in the scrum process. They are known as ceremonies and include:

Sprint Planning Meeting — This is where the team and product owner meet to discuss the planned project and determine what will happen during the sprint. The product owner outlines customer requirements that have been prioritized by business value — the product backlog. The team and product owner then decide which of these items will be moved into the sprint backlog. This section of the meeting should take no more than four hours. Another four hours are set aside

for the team to convert backlog items into actual tasks and work out how long these will take. At the end of this meeting, a detailed sprint plan with clear goals has been created.

Daily Scrum Meeting — These occur daily at the same time and should take no longer than 15 minutes. They are run by the scrummaster, and the whole team must attend. The daily scrum meeting provides a chance to assess progress and determine whether there are any problems that are slowing the project. Each member of the team should be answering these three questions during the meeting:

What have I achieved since yesterday?

What will I achieve before the next meeting?

What's stopping me from achieving what I want to achieve?

The answers to these questions give an indication of progress and highlight any problems that need to be solved. Nonteam members can attend the meeting but are not allowed to interrupt or comment. The reason for this restriction is often explained using the story of the chicken and pig planning to open a restaurant. "What shall we call it?" asks the pig. "Ham and eggs," replies the chicken. "No thanks," says the pig. "You're involved, but I'm committed." The team is committed to achieving a goal but should not be unduly influenced or constrained by interested parties.

Sprint Review Meeting — This takes place at the end of each sprint phase. The scrum team demonstrates its progress to its customers. These are supposed to be informal meetings and should take no more than four hours. In this meeting, the software that's been developed during the sprint is compared with the backlog. The backlog is then altered accordingly, with new items being added as priorities change. These backlog items are then assigned to the team, and the next sprint is planned.

Sprint Retrospective Meeting — These are also held at the end of each sprint. They occur after the sprint review meeting. They take no more than three hours and involve only the scrummaster and the scrum team. Their purpose is to review what went well during the sprint and look for ways to make improvements.

Scalability

Scrum has proven itself on small teams of 12 or less and scalable to projects of 50 or more. There are many considerations to successfully scale scrum. Chief among these are nonfunctional requirements and solution infrastructures. As project sizes increase, multiple scrum teams are created to manage complexity. Each team will have a subset of the overall project backlog. As nonfunctional requirements, such as performance or scalability, span all teams, these have to be agreed on and synchronized. This will require some upfront analysis to prioritize and communicate the requirements to all teams.

Business and system architectures need to be clearly defined to reduce team coupling and interdependencies. Larger projects also benefit from common development, and build and test environments. Each team has a daily meeting as per normal. The teams' scrummasters have daily meetings to share progress and resolve any team interaction issues. This requires careful logistics so that the teams do not trip over each other.

Is Scrum for You?

Successful adoption of scrum depends on many common agile success factors (see "Agile Success Factors" 148114). A number of specific pros and cons to scrum are outlined below.

Pros

- It is a well-documented and mature method with global adoption.
- Training and mentoring are readily available.
- It scales to multiple teams for larger projects.
- Tool support is available for metrics and reporting.
- It is proven to provide accurate estimates of schedule and effort.
- It complements the popular extreme programming method, with scrum providing the project management framework.
- It has been used within Capability Maturity Model Integrated organizations at Level 3.

Cons

- It provides project best practices but does not provide developer-level advice.
- It requires considerable change in business and IT relationships with regard to reporting and interaction.
- Plan-driven organizations require time and effort to move to empirical control-based methods such as scrum.
- Its use within regulated or safety-critical environments requires additional control.

Key Facts

- Jeff Sutherland and Ken Schwaber are credited with scrum development.
- Scrum is one of many agile methods that has taken ideas from manufacturing and applied them to software development.
- Scrum is focused on project management rather than development practices.

RECOMMENDED READING

"Agile Methods in Regulated Environments" 144396

"Agile Development: Fact or Fiction" 140892

"Agile Success Factors" 148114

REGIONAL HEADQUARTERS

Corporate Headquarters

56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters

Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters

Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters

Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters

Gartner do Brazil
Av. das Nações Unidas, 12551
9º andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509