

AgileLeanBooks

Scrum - a Bird's Eye View

How to lead product development, projects and work in
general in a complex world

Kurt B. Nielsen & Anne Due Broberg
AgileLeanHouse A/S, info@AgileLeanHouse.com
Lysholt Allé 6, DK-7100 Vejle, Danmark

Table of Contents

Introduction	3
The foundation	3
Where did Scrum and Agile come from?	3
A Timeline	4
The Cynefin Framework	6
To the Bird's Eye view then	7
The work is split into iterations	7
All requirements in a single prioritized list of deliverables	8
The players have distinct roles	9
The Product Owner	9
The Development Team	9
The Scrum Master	9
The rest of the organization	10
The Scrum activities or ceremony	10
Sprint Planning 1	10
Sprint Planning 2	11
Daily Scrum	11
Sprint Review	12
Sprint Retrospective	12
Product Backlog Refinement	13
Artifacts to improve visibility	13
The Product Backlog	14
The Sprint Backlog	15
The Impediment Backlog	15
The Product Burndown	15
The Sprint Burndown	16
Working in a Sprint	16
Definition of "done"	17
Summing up	17
Definition of terms	18

Copyright © 2017, AgileLeanHouse A/S. All rights reserved.

The content of this document is subject to change without notice and does not represent a commitment on the part of AgileLeanHouse A/S. AgileLeanHouse A/S makes no warranty of any kind with regard to this material.

AgileLeanHouse A/S shall not be liable for any errors contained herein or damages, including any loss of profits, or other incidental or consequential damages, arising out of your use of this written material.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information recording and retrieval systems, for any purpose, without the express written permission of AgileLeanHouse A/S.

All registered and unregistered trademarks used in this document are the exclusive property of their respective owners.

Document ID WP-Scrum-011
Author Kurt Nielsen

Filename
Revised

ScrumABirdsEyeView32
3.2 2017-10-21

Introduction

Scrum is a way of thinking and working - a way to wrestle results and motivation out of projects and undertakings in situations too complex for traditional planning. Scrum is one of the “Agile Methods” and they fit under the larger umbrella of “Lean Thinking”. Scrum is first and foremost defined through the document “The Scrum Guide” (SG), that is maintained by the fathers of Scrum Jeff Sutherland and Ken Schwaber, read more [here...](#)

Scrum came from the world of IT projects in the 1990s and rose to prominence in the new millennium as the most widely used of the Agile methods to manage projects. Scrum is sufficiently general in its foundation to be used in all sorts of projects and initiatives¹ - also outside the IT industry. Steve Denning, a thought leader in leadership, management and innovation, once said: *“If there was a Nobel Prize for management, and if there was any justice in the world, I believe that the prize would be awarded, among others, to Jeff Sutherland, Ken Schwaber and Mike Cohn for their contributions to the invention of Scrum.”* Scrum really is a revolution in leadership.

According to recent research², companies and teams that are using some form of Agile are consistently more successful, Scrum being the most popular with 85% using this. The year 2001 marked the new era: Ken Schwaber wrote his best selling book “Agile software development with Scrum” and the Agile Manifesto was signed by a group of influential software professionals. Since that year Scrum has spread like wildfire. There are now more than 430.000 certified Scrum Masters under the Scrum Alliance³ capable of running Scrum projects. As an example, many projects in the public sector in Norway are now required to follow Scrum principles.

Why use Scrum and Agile methods in initiatives, for example in projects and product development, then? First of all, there is solid empirical evidence documenting higher yield, lower cost, higher predictability and better stakeholder satisfaction. We also believe that it is beginning to dawn on many decision makers that it is impossible in today’s complex projects to come up with the perfect upfront plan. It is not even possible for us to attain perfect specifications; technology changes every day and external dependencies fluctuate as well. We are simply not in an ordered domain in most of our non-trivial projects - we are in the complex domain, and totally different measures are called for.

The foundation

Where did Scrum and Agile come from?

Scrum is often called Agile project management, and Agile is roughly defined as anything work related that is focused on human interactions, results, collaboration with customers and the concept of adapting to change⁴. In a wider perspective, it is legitimate to see Scrum and Agile as part of the greater family of Lean, focusing on customer value, reducing waste and respect for people.

However, there is even more to keep in mind when trying to assess the foundation of Scrum and Agile. Complexity Science tries to describe how some understanding of complex adaptive systems can be obtained. By definition Complexity Science is complex and hard to contain in a few simple buzzwords. We can start is to look at the Cynefin (pron.: [/ˈkʌnɪvɪn/](#))

¹ We shall use the word “Initiatives” to cover projects, development of products and services plus work in general.

² 2016/17 research, Prof. Dr. Ayelt Komus, University of Applied Sciences, Koblenz

³ See <https://www.scrumalliance.org>

⁴ See the Agile manifesto at <http://agilemanifesto.org/>

framework, formulated by Dave Snowden⁵. Cynefin has proven very useful in understanding how to deal with the very different situations we experience when managing projects, initiatives and organizations in general.

In this perspective, we can begin to understand why Scrum works so well.

A Timeline

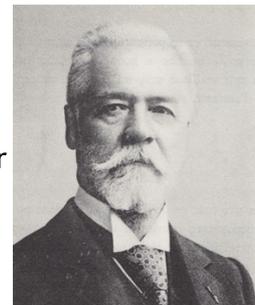
Fredric Winslow Taylor (USA 1856-1916) was the first to apply systematic study to the subject of work and he became the father of the so-called Scientific Management in the beginning of the 20th century. His book from 1911 on the subject (“Principles of Scientific Management”), marked the beginning of an era with focus on efficiency, as most notably can be seen in Henry Ford and his assembly line for the model T.



Taylor’s ideas gave birth to concepts such as the Gantt chart and budgeting. Taylor, however, being deeply rooted in industrial production, was completely convinced that every piece of work could be reduced to a set of simple repeatable tasks, which could be measured and studied. It would always be possible for the expert to find a best way of doing things (best practice).

Taylor’s view of the average human being was not attractive however. He basically considered the average worker too stupid and lazy to have influence on his own work; experts had to plan for him and drive him with a combination of carrot and stick to do something, execute tasks. This concept is still with us today.

Henri Fayol. (France 1841-1925) was a mining engineer who ended up being in charge of a large mining company. Here he developed over the years a set of 14 principles that he was convinced were absolutely foundational to running a large company. In 1916 he published the book *Administration Industrielle et Générale*, one of the first works with a clear theory for management and still one of the most elaborate.



Although Fayol is not generally well known, his ideas about administration, line of command etc. has had a profound influence over the last 100 years. His view of management as those delegate, coordinate, supervise and control work is still with us.

Alfred Pritchard Sloan, Jr. (USA 1875-1966) was a long-time president, chairman, and CEO of General Motors Corporation. He reorganized the company in a way that became the template for almost all companies in the 20th century. He divided General Motors into separate autonomous divisions that were subject only to financial, budgetary and policy controls from a small central staff. Sloan also introduced a systematic, detailed strategic planning procedures for the company’s divisions, the first CEO ever to do that. His focus on budgets and minute upfront planning is still with us today.



Walter A. Shewhart (USA 1891-1967) was a physician, engineer and a statistician, he described as the first ever the possibilities of statistical quality control. He did this during his work at Western Electric in the 1920s. Shewhart’s work showed the necessity of reducing variation in production processes and work in general. In this way he fathered the first ideas about constant improvement. Shewhart created the first versions of control charts, that later have been widely used to monitor



⁵ For an introduction watch this video presentation <http://www.youtube.com/watch?v=N7oz366X0-8>

all sorts of processes, in order to determine if the results are likely to be caused by random variation in the process or specific causes that can be hunted down.

W. Edwards Deming (USA 1900-1993) has contributed to a total focus on quality and involving the whole person in the work process. He was in opposition to Taylor, Ford and Sloan. He contributed to the American war effort during the Second World War by helping create and teach the quality standards crucial to the precision manufacturing of America's war equipment. Later, in the 1950s, Deming was instrumental in Japan's transformation into a first class industrial nation. He learned a lot from his mentor Shewhart in the 1930s and he can in many ways be considered to be the father of Lean.



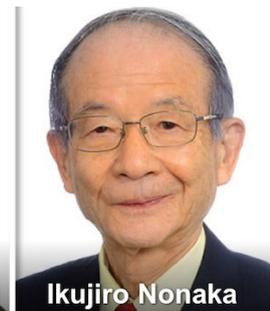
In the 1980s Toyota's operating models in production, product development and supply chain management became known in the Western world, and this spawned a change in ways of organizing work in the years to come. We know this now as "Lean".

Tom Gilb is an American software consultant living in Norway. In the 1980s he came into the software world with the book "Principles of Software Engineering Management"⁶ from 1988. In the book Gilb describes, among other things, an iterative development- and delivery principle called Evo, which still remains valid today.



If Deming is the great-grandfather of Scrum as we know it today, Gilb would for sure be the grandfather. He is still active as a coach and consultant today.

Hiroataka Takeuchi and **Ikujiro Nonaka** published In 1986 a report in the Harvard Business Review, it was called "The new, new product development game". It described a new parallel way of finding solutions, which the authors compared to Rugby: it described a process characterized by self-organizing project teams, overlapping development, multi-learning and "subtle control". They introduced the word "Scrum" used in Rugby.



Ken Schwaber and **Jeff Sutherland** picked up on this line of thinking along with influences from process theory and their "mission" thinking from their military background. They, the "fathers", developed and formulated Scrum through the 1990s. The word Scrum was chosen from the rugby vocabulary that Takeuchi and Nonaka had introduced. Schwaber and Sutherland linked their understanding of Scrum strongly to empiric process control, which is often applied when the underlying mechanisms of a process are inadequately understood for working after a prescription. Other streams also influenced the concept (most notably Extreme Programming as formulated by Ken Beck in 1999⁷).

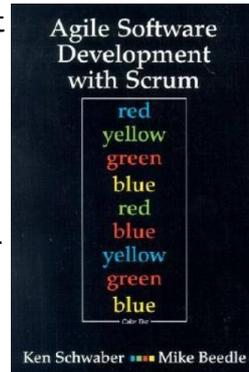


⁶ Principles of Software Engineering Management, 1988. ISBN 0-201-19246-2 (19th printing).

⁷ Extreme Programming Explained, Ken Beck 1999

Ken Schwaber wrote a book in 2001 called “Agile software development with Scrum”, and that really popularized Scrum and propelled the concept into wide acceptance in the IT and software industry.

The last 15 years has seen a wide acceptance of Scrum and other agile practices. Several things have been adjusted in the concept and new elements have also been introduced. It has remained largely a pragmatic approach though, which is good; there is something in it you can use immediately with good results. Scrum has been defined as ‘*common sense made visible and applied systematically*’.



However, until recently, Scrum has remained largely un-explained theoretically, apart from the connection with empiric process control. But now - through the work of **Dave Snowden** and his Cynefin framework - we are able to offer a much better explanation of why, we do, what we do in Scrum. Scrum has over the years developed to be a pretty good match for working in the transitions between the complex and the complicated domains, enabling us to apply the right methods to the situation at hand. The Cynefin framework in turn gives us the theoretical foundation to understand all this.



The Cynefin Framework

The Cynefin framework was developed by Dave Snowden & Cynthia Kurtz from 1999 to 2005. It is used to describe problems, situations, systems and leadership. By offering a thinking-model, a sense-making model so to speak, it explores how we deal with different types of challenges.

The Cynefin framework has four domains and “Disorder” in the middle:

- **Obvious (Simple)**, in which the relationship between cause and effect is obvious to any reasonable person. The approach is to “*Sense - Categorize - Respond*”. We can apply best practice.
- **Complicated**, in which the relationship between cause and effect exists but requires analysis or some other form of investigation and/or the application of expert knowledge. The approach is to “*Sense - Analyze - Respond*”. We can apply good practice depending on the expertise at hand.
- **Complex**, in which the relationship between cause and effect can only be perceived partially or in retrospect, but it cannot be completely understood in advance. Many different actors modify the system and each other. The approach is to “*Probe - Sense - Respond*”, we have to try things to make sense of it all. We can sense emergent practice.
- **Chaotic**, in which there is no visible relationship between cause and effect at systems level. The approach is to “*Act - Sense - Respond*”. We can discover novel practice.



- **Disorder**, which is the state of not knowing which domain you are in. In this state people will assume to be in the domain which they are most used to and revert to making decisions based more on past experience and comfort.

The boundary between simple and chaotic looks different than the other boundaries in the model, because it signals a change which could be catastrophic: Oversimplifying the situation, relying on procedures and best practices in circumstances with accelerated change and great uncertainties leads to complacency, which can cause catastrophic failure in a crisis.

In many of our projects and endeavors we are in the Complex domain and should act accordingly. As leaders in the complex domain we need to (according to Snowden):

- Probe, sense, respond. We do “safe-to-fail” experiments, not “fail-safe” designs.
- Create environments and experiments that allow patterns to emerge.
- Increase levels of interaction and communication.
- Use methods that can help generate ideas:
 - Open up discussion (as through large group methods).
 - Set barriers and constraints.
 - Stimulate attractors - things that produce attractive results.
 - Encourage dissent and diversity.
 - Manage starting conditions and monitor for emergence.

To the Bird's Eye view then

While there are many different agile approaches, we will focus on Scrum, as it is mainly concerned with the high level principles across domains. We will then leave the professional skills to be defined within the domain that applies. For example in an IT project we might expect the Team to use principles from Extreme Programming to achieve speed and quality. Scrum is a well-documented method that addresses the constraints we apply to the situation; we do not have a detailed recipe to follow. In this chapter we sweep across the main components of working the Scrum way from a fairly high level.

The work is split into iterations

In Scrum we realize that we are in a world of changing circumstances: priorities may change, users change their minds, technology is different from what we assumed, suppliers may let us down - all in all projects carry all the characteristics of a complex domain.

In order to get something done, we split the whole project into a series of iterations called Sprints of normally two to four weeks length⁸. We ask “what are the most important things we could spend our next Sprint on?”, and then we do it. At least we have taken the complexity out of “what to do” for a short time.



⁸ The Scrum Guide now says that 1 week is OK, we don't recommend that.

This is actually an implementation of the PDSA (Plan-Do-Study-Act) circle⁹ introduced by W Edwards Deming, although often used to ensure quality; its underlying methodology is valid here as well.

Another way to explain this is that we use a linear approximation to a non-linear process. In the complex domain, there is a non-linear cause and effect relationship, but we decide to regard it as linear for a short period of time. We also do everything, we can, to isolate the Team against changes during the Sprint, so that we with reasonable precision can predict outcome.

It is like navigating a submarine: Periscope up! Where do we want to go for the next period? Decision, periscope down! Follow course for the set time period. Periscope up! How does the world look now? Where do we want to go next? You get the drift.

The sprint creates a boundary in the complex space; within that created arena the Team can self-organize, find solutions and enter flow¹⁰. Based on experience, the Sprint should be between two and four weeks. Shorter Sprints do not really allow the Team any slack in finding solutions and there seems to be a feeling of having to finish before getting started. Longer Sprints tend to relax the focus on the goal too much; it is easy to feel that the Sprint-end is very far away, so we end up having a very un-even speed during the Sprint with poor predictability and probably poor quality as the result.

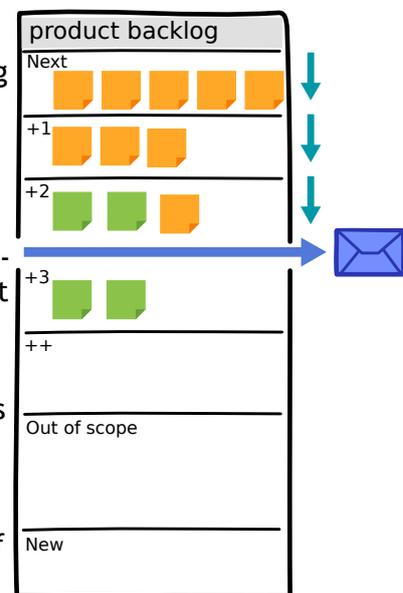
All requirements in a single prioritized list of deliverables

Everything, we want to have delivered, needs to have some description. This being Agile and Scrum, of course, details may change and indeed new requirements may surface during the project.

We collect all these requirements in a sorted list with the ones to work on first at the top. We call this list the **Product Backlog**. Exactly one person, the Product Owner, has the responsibility for this list, more about this in a moment. The requirements are formulated using a standard template, so that they can be compared and prioritized against each other. These are called **Product Backlog Items**.

These items are often formulated as User Stories¹¹, narratives of much the same nature as used in Cynefin. The Team also self-assesses (estimates) the size of the work related to a Product Backlog Item. Other people (users, customers, the business) will do the same with the value of an item. Some of the items may be complex, some complicated and some simple/obvious.

The Product Backlog is also a deliberately chosen boundary; everything we want to have to done goes here. Some things will be experiments, some analysis, some ordinary delivery requirements and probably a few relatively unrelated things, that just need to be done. Everybody knows where to look for what to do and what the current prioritization is, the know who to talk to if they disagree with the prioritization, the Product Owner. The Product Backlog is public for all stakeholders.



⁹ See here: <https://www.deming.org/theman/theories/pdsacycle>

¹⁰ Flow, a psychological condition, read more here: [https://en.wikipedia.org/wiki/Flow_\(psychology\)](https://en.wikipedia.org/wiki/Flow_(psychology))

¹¹ Although Scrum does not require the use of this format, it is just a good practice.

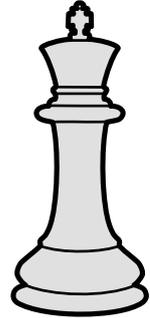
The players have distinct roles

The next boundary that is defined in Scrum is the set of distinct roles that the different participants play. The set of people committed to produce results through the project are collectively called **The Scrum Team**. They consist of the **Product Owner**, the **Development Team (Team)** and the **Scrum Master**.

The Product Owner

One person takes responsibility for prioritizing what to work on; he is called the **Product Owner (PO)**. In the wider perspective he takes responsibility for Business Value generated in the project and the return on investment (ROI).

The PO understands the domain of the customers and users and also takes responsibility for explaining this to the Team working on the deliverables. The Team has exactly one point of reference as to “what we build” and “when”, that is the PO.

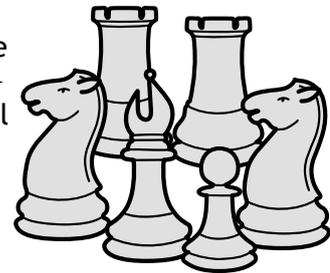


The PO operates in the **strategic** area navigating the domain with complex and complicated items and he plans the **tactical** Sprints together with the Team. The PO constantly works with the Product Backlog, makes improvements on explanations, gathers new ideas, acquires acceptance criteria, listens for signals from users, re-prioritizes etc.

The Development Team

A small team (5 to 9 individuals¹²) takes responsibility for doing the work in the project; we call this the **Development Team** or short - the **Team**. The Team is self-organizing, cross-functional and has all the necessary skills to actually produce the deliverables as described in the Product Backlog.

Ideally there is no dependencies to anybody or anything outside the Team. Ideally the Team does not have other jobs to do than those present on the Product Backlog.



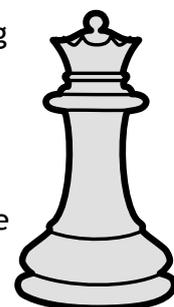
The authority to find the right solution “how to build” the deliverables rests with the Team. They have the necessary authority, resources and commitment to assume this responsibility. The Team chooses in dialogue with and based on the priority of the Product Owner the set of Product Backlog Items that they can commit to in the following Sprint. After this they are left to do the job in the Sprint. It is also the Team’s responsibility to make their progress and potential impediments visible to everybody.

The Team works with the Product Owner to raise the understanding of Product Backlog Items, such as finding acceptance criteria. The Team estimates the size of work associated with Product Backlog Items and helps the PO to decompose large PBLs.

The Scrum Master

One person takes responsibility for governing the Scrum process, implementing improvements, removing impediments and protecting the Team and the Product Owner. He is called the **Scrum Master (SM)**. He is not the usual project manager or team leader; instead he is a servant leader.

The Scrum Master is focused on achieving a spirit of constant improvement, so he has to be deeply convinced of the values of Scrum. Often he will be the one to work with the whole surrounding organization and explain why and how Scrum is used.



¹² The Scrum Guide now says 3 people are OK, that may be so, but you lose something.

The Scrum Master adds value to the project by always observing and listening for weak signals, indicating either opportunity or potential danger. In a way he is the one, who is mostly exposed to the complex domain, constantly living and breathing in it, making sure that the entire Team doesn't slip into mistakenly assumed simplicity, where this does not exist. He tries to create room in the Sprint - in time and space - for the Team to be able to find and create solutions and get into flow; this is partly done by protecting the Team from the constant changes and interruptions from the surroundings.

The Scrum Master is like the coach of a sports team, like the conductor of the orchestra or like the sheep-dog watching over the Sheep - at all times serving the Team and the Product Owner.

The rest of the organization

The Scrum Team interacts with stakeholders and with management. Stakeholders typically include customers that in some way pay for the deliverables and users that are the consumers of what is produced.

Management is involved in providing the necessary tools and resources for the Scrum Team to use and in removing obstacles, typically on the initiative of the ScrumMaster.

Everybody can place items on the Product Backlog, look at the public artifacts and participate in the public Scrum activities.



The Scrum activities or ceremony

Scrum defines a series of regular activities or meetings, each with their own specific purpose, this sets another set of boundaries, to make sure that we stick to the process and stay on the alert observing for signals of change. Keeping a rhythm to the work also helps everybody get into a good habit and thus develop a culture that matches the complex domain with frequent inspections to set the course.

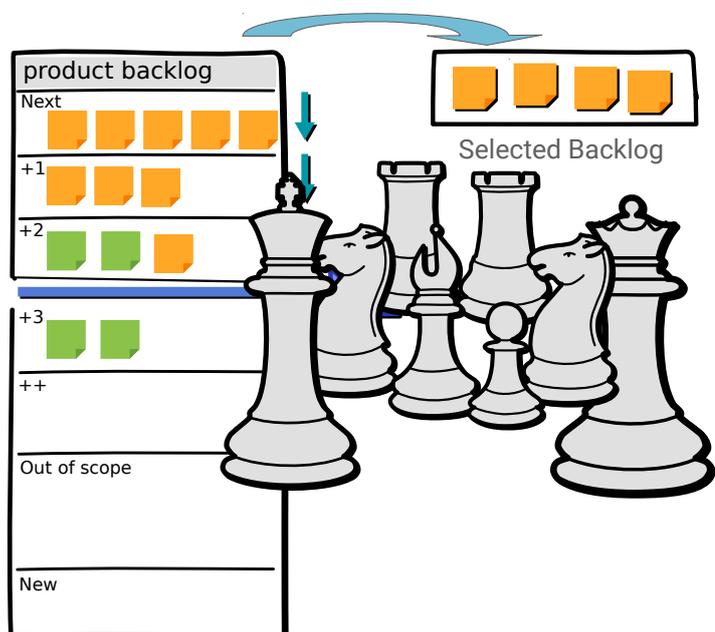
Meetings in Scrum are **time-boxed** for a couple of reasons. Without time-box control meetings can go on for ever and become very unproductive, the slight pressure of the time-limit also helps to achieve a result and allows people to move on to their real work. Every attendant then knows what to expect and can plan other activities accordingly. There has to be that slight sense of urgency.

The Scrum Master is present at all meetings and facilitate these.

Sprint Planning 1

At the very beginning of the Sprint, the Team and the Product Owner meet to select the optimal set of Product Backlog Items for the Sprint in question. This is a time-boxed meeting, approximately 1 hour for each week of Sprint length.

Before entering the meeting the Product Owner has ordered and prioritized the Product Backlog and relevant items are estimated by the Team. The Product Owner has also



formulated a proposed Sprint Goal, expressing the essence of what will be achieved in the Sprint. The Team has beforehand assessed their capacity and has a qualified estimate of the Velocity for the Sprint. So they have a pretty good idea of how much they can do.

The Product Owner presents the relevant items. If necessary more people with special insight are brought in. The Team asks questions, there is dialogue and a building up of common understanding. New knowledge may be harvested and change in estimates and prioritization may occur.

In the end, the best possible set of Product Backlog Items to implement is agreed to and selected for the Sprint. At the end of this meeting we have crossed over from the strategic area to the tactical area, from “what to do” to “how to do it”.

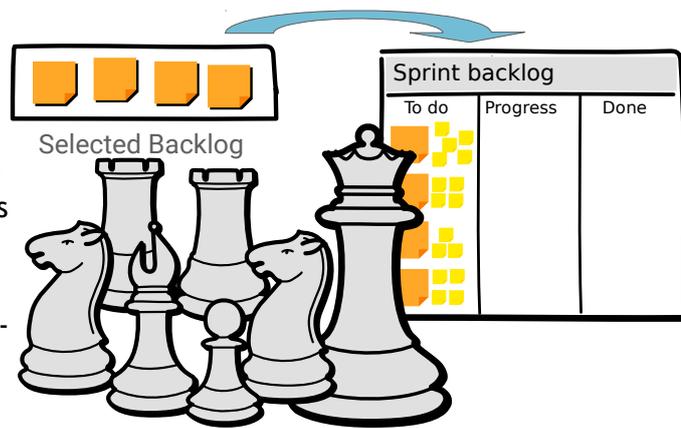
Sprint Planning 2

Then the Team then continues on their own with the second half of the Sprint Planning called **Sprint Planning 2**; the PO stays on call. Some Teams decide to include the PO in Sprint Planning 2. This meeting is timeboxed to about the same size as the Sprint Planning 1.

The Team now finalizes analysis and design of each Product Backlog Item and breaks it further down into smaller pieces of work, typically formulated as **Tasks**. The principle is that a Task should only last maximum one day and then it should be finished.

If new questions or doubts arise, the Product Owner is called back to enlighten the Team further. The Team has assessed the items before, both in Sprint Planning #1 and in Product Backlog Refinement. If new knowledge is harvested, the selected Product Backlog Items may still change. The Team builds a **Sprint Backlog** of work to do.

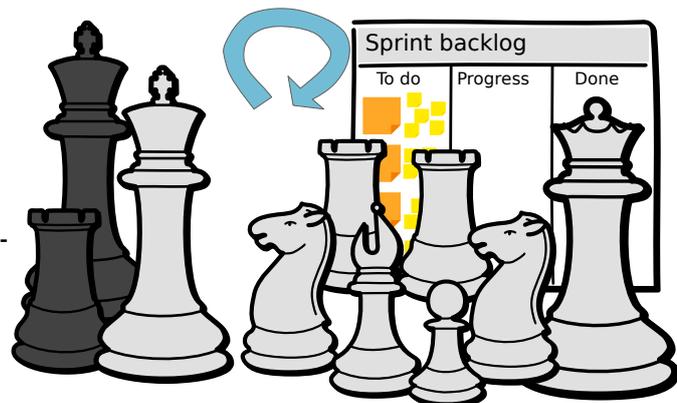
When the meeting is over, we have a Sprint Goal, a Commitment - sometimes called a forecast - as to what the Team honestly believes, it can deliver in the Sprint. The Team commits to do their best to accomplish this goal, a set of Product Backlog Items and a Sprint Goal. The Team also has a pretty good idea of how, it is to be done. This is then made public as the Sprint Backlog.



Daily Scrum

During the Sprint, the Team and the Scrum Master meet every day at the same place and time to synchronize tactical activities and follow up on progress. This is called **Daily Scrum**. Everybody (including the Scrum Master) answers three questions:

- What have I accomplished since yesterday?
- What will I accomplish until tomorrow?
- Are there any impediments preventing my optimal work?



The Daily Scrum is time-boxed to 15 minutes and is public; people outside the Team can come; but they do not speak. If they have comments and questions they must address the ScrumMaster after the meeting.

During Daily Scrum the Team and the ScrumMaster may discover the need for further discussions; extra meetings are quickly scheduled, issues are not dealt with in the Daily Scrum.

New impediments and obstacles are recorded by the Scrum Master on the Impediment Backlog and immediately after Daily Scrum he attacks the issues.

The Scrum Alliance recommends that Product Owners participate in Daily Scrum and speak as well. Some Teams find this helpful.

Sprint Review

At the end of the Sprint a **Sprint Review** where the results of the Sprint are presented. The Team and the Product Owner are present, all stakeholders are invited and everybody can come. Only the Product Backlog Items that were finished in the Sprint are presented and demonstrated by the Team. Everybody is encouraged to give feedback.



The Product Owner is of course present and formally approves the delivered Product Backlog Items or in case of disapproval records new Product Backlog Items describing the needed correction. New input and ideas may come up; the Product Owner records these for the Product Backlog.

The Sprint Review is time-boxed as any other Scrum meeting, the recommended length is the same as the Sprint Planning meetings. It is however often it is kept on the short side (like 90 minutes) to accommodate other people in the organization with busy schedules. This meeting closes the PDSA loop with respect to the **Product** being delivered. The meeting also makes sense seen in the light of Cynefin as an invitation to many with diverse background to provide feedback, not just the experts.

Sprint Retrospective

At the very end of the Sprint, the Team and The Scrum Master get together to share experiences from the Sprint and reflect on potential improvements, this is called the **Sprint Retrospective**.

The Scrum Alliance recommends that the Product Owner be included in this meeting. Some teams find this helpful.



The meeting is timeboxed as the other meetings. However many teams find it useful with a shorter meeting, say an hour.

There are many forms of the Retrospective that the Team and the ScrumMaster can choose from. What is important to achieve is:

- The Team and the Scrum Master share the facts of the experiences of the Sprint.

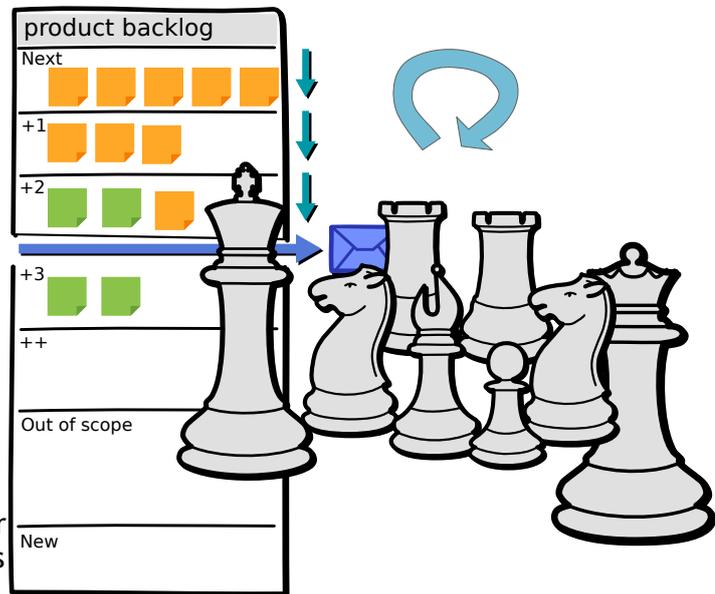
- They reflect on the meaning of these experiences and pass judgment on what is positive and what could be improved.
- They prioritize what they want to try to improve and make a decision to act. What will the Team do, What will the Product Owner do and what will the Scrum Master do.

This meeting closes the PDSA loop with respect to how the work is being executed, the process. Again the meeting is in line with the thinking of Cynefin. We recognize that this is more than a complicated expert domain. By letting the people involved self-assess greater insight is achieved.

Product Backlog Refinement

There is yet another important meeting, the **Product Backlog Refinement**. The Product Backlog needs to be thoroughly understood, prioritized and relevant items estimated before entering Sprint Planning. To make sure this is indeed happening, we institute this meeting, typically held in the middle of the current Sprint with focus on the next Sprint.

It is a preparation for the Sprint Planning 1 meeting, so it looks almost the same. The Product Owner and the Team are present. The Product Owner presents the items, he has selected as candidates for the next Sprint along with new items that have surfaced or perhaps old items, where circumstances for some reason or another have changed. Again there is dialogue and building up of common understanding.



The participants may break large Product Backlog Items down, refine descriptions, capture acceptance criteria and then estimation of cost.

The difference between Product Backlog Refinement and a Sprint Planning #1 is that instead of selecting items, the items are simply estimated for cost (typically for size of work). It is important that the estimation is performed by the people, who in the end will have to deliver - the Team.

Product Backlog Refinement is of great value to the Product Owner, as it enables him to improve the quality of his prioritization. The activity gives him the best possible understanding of the Items and an estimate of cost. After the Product Backlog Refinement he still has time to prepare his prioritization for the next Sprint.

Artifacts to improve visibility

Artifacts are “evidence of human activity” and is used in Scrum about the lists, boards, graphs and systems, being used to visualize and manage the whole Scrum process. They are “information radiators”.

An empirical process control like Scrum is supported by three pillars: **transparency, inspection and adaptation**. The transparency part is achieved partly by making select activities and meetings public and holding these regularly, but also through the information radiators, which we call Artifacts. The Artifacts are easy to understand and they are self documenting, making it possible for even the innocent by-stander to derive meaning from them.

In Scrum we use a number of these Artifacts, in Cynefin terms they also provide input to the “sense-making” process, what do our observations mean?

The Product Backlog

The **Product Backlog** is an ordered list of Product Backlog Items, individual deliverables that we want, but do not have yet. The items at the top are considered more desirable to start working on early compared to those further down the list. This ordering is made by the Product Owner, who owns the Product Backlog; only he changes the ordering of items, although anybody can add to the Backlog. At any time, the backlog is a snapshot of the current plan as defined by the Product Owner.

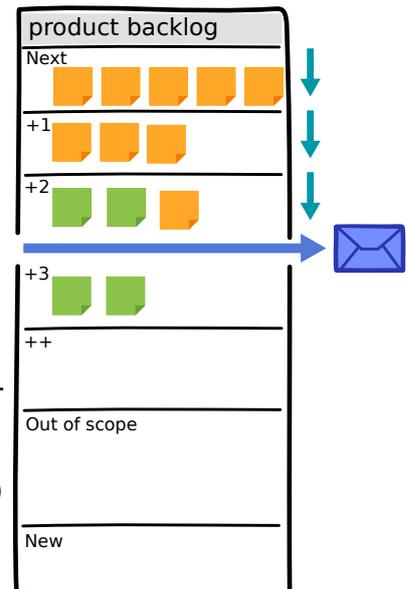
The Backlog will often be shown with sections for different releases or phases and within them different planned Sprints. There is typically also an out-of-scope section (stuff the Product Owner does not think will make it within time and budget) and one for new ideas or requirements.

The Product Owner constantly works on refining and prioritizing the Product Backlog. This is called “refining the Backlog”. Sometimes he solicits help from the Team to do this refining, the Estimation meeting is one such common Backlog refining effort. Sometimes he has others in the organization helping him with this.

Each Product Backlog Item contains enough information for the Product Owner to plan and the Team to implement. The information is meant to be headlines and reminders of having detailed conversations. Backlog Items often contain information such as:

- An ID - some people dislike this idea of a unique ID
- A title or a name
- An identification of a major area, typically called a “Theme”
- Estimate of size of work, often including uncertainty in the form of a 50% and a 90% estimate.
- Estimate of Business Value, often including uncertainty.
- Estimate of monetary cost.
- Origin of item, who and when?
- Reference to more information.
- Perhaps an estimate of Cynefin Order and Kano assessment¹³.
- Description framed as a User Story: “*As such-and-such a user, I would like to do this-and-that, in order to achieve jada-jada-jada.*”
- Extra information about detail specifications, typically recorded as acceptance criteria: “*In such-and-such-a-situation, if event occurs, the result should be so-and-so!*”

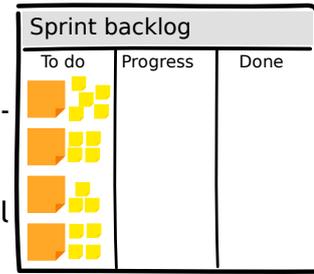
The Product Backlog shows the current plan and prioritization to everybody interested. It can be a physical board with post-it notes or index-cards, it can be a spreadsheet or it can be implemented in a dedicated system.



¹³ See here: https://en.wikipedia.org/wiki/Kano_model

The Sprint Backlog

When the Team has planned a Sprint they represent that in the **Sprint Backlog**, an artifact showing the planned deliverables (Product Backlog Items) and a plan for the work in the Sprint. This is often implemented as a Task Board with post-it notes in the Team's room. It is highly visible and documents the Team's progress in real time.



Each Product backlog Item is broken down into smaller pieces of work, typically tasks. Each piece of work can be in different states, typically as a minimum we have “Waiting”; “Progress”, “Blocked” or “Done”, where “Blocked” means that the Team is waiting for something outside the Team. Work is normally broken down until each piece should only be in “Progress” one day.

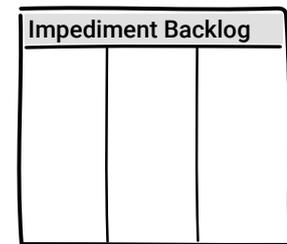
It is common to record unplanned activities often using different colored post-its, as these often will be present the most serious obstacles to the Team's progress and constant improvement. They will need to be addressed and making them highly visible makes the organization react to them. They will often be discussed on the Sprint Retrospective.

The Sprint Backlog may reside in an IT system, if the Team is not co-located, but a physical Task Board is considered by most Teams to be far superior; there is constant visibility.

The Sprint Backlog is owned and maintained by the Team, and it is together with the Daily Scrum the way the Team shows to surrounding organization the progress being made and it is the way the Team generates trust. The Team updates the Sprint Backlog at least once a day, so that it is correct during Daily Scrum.

The Impediment Backlog

It is common to have an Impediment Backlog. It is almost like the Sprint Backlog, but instead of displaying the work related to the product everyone in the Team works on, it displays the things, the ScrumMaster works on, in order to improve working conditions for the Team, so that they can reach their maximum potential.

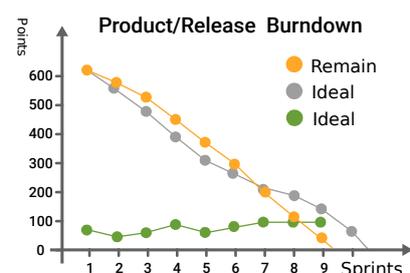


The Impediment Backlog is a delicate instrument. By its sheer public existence it provokes action: making impediments visible will often start the process of removing them. Nobody wants to be on the Impediment Backlog. However, this also can provoke antagonism and conflict. The ScrumMaster has to handle this situation with finesse.

The Product Burndown

Stakeholders and the Product Owner often want to monitor the progress of the whole project and be able to make projections about, when the project will finish, or alternatively how much can be achieved within a certain time frame.

It is common practice in Scrum to use a **Product Burndown** chart that shows how much work has been achieved, and of course how much is left (based on the estimates) before a certain goal is reached.



The Product Burndown records for each Sprint how much work is left. From this the Team's Velocity can also be monitored: how much is the Team capable of doing per Sprint? If we implement changes to the Team's work environment, it is important to monitor if it indeed improves the Velocity.

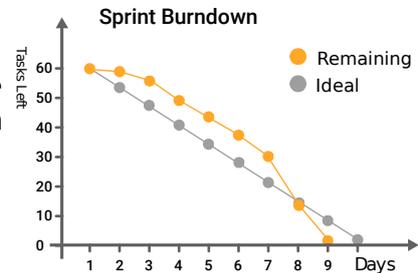
Given a qualified estimate of the Team's Velocity, predictions about estimated time of arrival can be made, typically as an interval.

It can also be recorded on the Product Burndown, how much Business Value has been harvested, how much risk is in the estimates and so forth.

It is common to display new items added to or items removed from the Product Backlog on the Product Burndown. In the same way, it must be noted on the burn-down, if major re-estimation has changed the perspective, which should never be confused with change in the Team's velocity. Sometimes the focus is only on the next phase or release, then we call this artifact the **Phase or Release Burndown**.

The Sprint Burndown

If the Sprint is long (like four weeks), the Team is big or the general complexity of the backlog items is rather high, then there may be a need for a **Sprint Burndown** chart to monitor progress within the Sprint. If the Team is uncomfortable about staying in control of progress or the Sprint often misses the goal; then you need the extra visibility and control of the Sprint Burn-down.



The Sprint Burn-down records the number of Tasks not “Done” every day in the Sprint. It is then possible to make predictions about whether the Team will complete everything planned for the Sprint. This is based on assumptions of the speed with which the Team executes tasks.

It is common to display new unexpected tasks added to or tasks removed from the Sprint, to prevent obscuring the progress by unplanned events.

Working in a Sprint

So what is it like to work in Sprints? What happens on a normal day?

The Team self-organizes with focus on getting Product Backlog Items moved as quickly as possible to “Done”. Once work on an item has been started, the focus is on getting it “Done”; work in progress should be limited as much as possible. The Team members help each other out, some switch roles temporarily to assist others, all use common sense. The Sprint Backlog is constantly updated when Tasks or Product Backlog Items shift status. The Daily Scrum is held to synchronize and keep the Team together. They focus on how to get the committed Product Backlog Items done and meet the Sprint Goal.

The Scrum Master is busy working with the rest of the organization on removing any impediments found or implementing any planned improvements in work conditions. The Scrum Master protects the Team from interruptions and acts as a buffer, while the Team is in Sprint. Any contact from outside goes to him. He also serves the Product Owner and tries to remove anything blocking his meaningful work with the Product Backlog.

The Product Owner assists the Team in its tactical execution and answers questions during the Sprint when the Team asks. Meanwhile, he is busy improving the Backlog and digging out business value. The Product Owner also works on clarifying not only specifications, but also acceptance criteria, risk and any other condition that might affect the prioritization for the next Sprint. The Product Owner plans for the next Sprint, and has estimation performed by the Team. He focuses on longer term planning, leaving the current tactical Sprint to the Team to sort out.

Everybody else with an interest in the project follow progress, by inspecting the Artifacts and showing up for Daily Scrum. If they have comments to the backlog, new ideas or different views on valuation, estimates or specifications, they direct this to the Product Owner. If they have comments to the execution of the current Sprint, they go to the Scrum Master, and he takes it further to the Team or arranges a meeting, if necessary.

Unplanned work that cannot be deferred to later Sprints should be avoided, but happens in real life. This breaks the plan. The Team solves these issues and records them as unplanned work on the Sprint Backlog for all to see.

Should the Team reach the conclusion during the Sprint that they cannot finish everything in the Sprint, they first see if there are alternative, easier ways to accomplish, what they set out to do. If this is not possible, they contact the Product Owner straight away, to discuss which of the remaining Product Backlog Items could be put back on the Product Backlog to reduce the work in the Sprint. This could happen, if unplanned work hit the Team, or if they discover that some things are harder, than they had thought during Sprint Planning. Of course, the Team also contacts the Product Owner for more Product Backlog Items, should the - slightly less likely - situation occur, where they find they will finish the Sprint early.

If the Sprint Goal becomes irrelevant, the Product Owner can abort the Sprint. That could happen if the customer goes broke or the Team is interrupted so often that they are nowhere near to meet the Sprint Goal. This rarely happens, and should of course be avoided, at it normally creates a lot of friction and conflict.

At the end of a Sprint the has the be true Product Increment an increment of the result wanted for the project or initiative. Only completely done Product Backlog Items count towards the result.

Definition of “done”

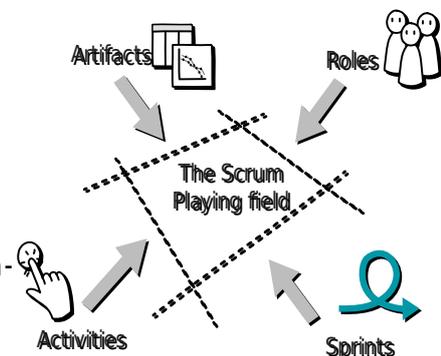
When a Product backlog Item is declared “Done”, all parties must agree on what Done means. As an absolute minimum, it must mean that the Team is not aware of any outstanding work that ought to have been executed. It is good practice to agree specifically on what qualities a “done” item should posses. This will include standards for the work is done, various standards to obey and various ways of verification.

This definition of Done, can come from the organization, providing certain standards. It is typical that the Team will produce various improvement to this definition in order to be even better, this is normally discussed during Sprint Retrospective.

Summing up

The Scrum framework sets some boundaries within the typically complex environment of a project. This assists in making sense of the environment, getting work done and achieving results.

- Work is done in iterations, with constant following up and monitoring in order to keep the course and to constantly improve.
- Deliveries of totally Done Product Backlog Items with feedback must be frequent, ensuring that the right things are delivered.
- Clear lines of responsibility must exist, making it possible for everyone to know what to expect and who to approach in different situations.
- The focus is on visibility, visibility and visibility. This improves accountability and gives insight enabling one to act quickly.
- The process must always be focused on value (the results delivered to the customer) and on constant improvement.



Addendums

Definition of terms

Term-EN	Abbr.	Description
Acceptance Test		A description of the Done criteria for a User Story, for example using the following template: "Given a--certain-context and some-more-context. When an-event Then an-outcome and another-outcome."
Acceptance Criteria		The same as Done Criteria
Artifacts		The tools used to visualize what to deliver, prioritization, tasks to do, impediments and progress. The term normally refers to the Product Backlog, the Sprint Backlog and the Impediment Backlog, but can also refer to the Product Burn-down and the Sprint Burn-down, if these are used.
Business Value		The quantified value that the Product Owner (on behalf of the Customer) expects to reap from getting a particular Product Backlog Item.
Constraints		Conditions for the project that are defined by the Product Owner (on behalf of the Customer). Any solution that is developed during the course of the project must be within the definition of the collection of constraints. This often includes performance, platforms, usability, documentation etc.
Cross-functional Team		A team where the members together possess all the necessary skills and competences to complete the Product Backlog Items in the Project
Customer		A secondary role in Scrum: the person who purchases the Project and pays for it. The Customer naturally wants maximum value and return on investment.
Daily Scrum		A daily, 15 minute, standup meeting held in the same place and on the same time everyday, where the Team members and the Scrum Master synchronize. Each person answers the three questions: what have you accomplished since yesterday? What do you plan to accomplish until tomorrow? Are there any impediments preventing you from working optimally?
Delivery-In		A milestone in the Time-line, specifying that at this time we plan to receive certain deliveries IN to the project, for example from other teams or suppliers.
Delivery-Out		A milestone in the Time-line, specifying that at this time we plan with certain deliveries OUT of the project, for example to other teams or to suppliers.
Development Team		The small cross-functional team that actually produces Product Backlog Items, this term is defined by Ken Schwaber and Jeff Sutherland. Also often called just the Team.
Done		Finished, finito, no-more-work. The state of a completed Task, piece of work or Product Backlog Item. The Team declares with confidence that there is no more work to be done according to Done-criteria, general quality levels and other constraints for the project.
Done Criteria		Those conditions that must exist for a certain Product Backlog Item for the Team to declare it Done. This includes user acceptance criteria.
Entire Scrum Team		All people having responsibilities and commitments in the Scrum process: the Product Owner, the Team and the Scrum Master.
Epic		An Epic is a large User Story, typically one that later on will be broken down into appropriately sized User Stories.
Estimate		Estimate of the size of work of a Product Backlog Item. Estimates are produced by

		the Team and used by the Product Owner to order and prioritize the Product Backlog.
Estimation meeting		A meeting, typically in the middle of a Sprint, where the Team and the Product Owner get together in order to estimate in detail relevant Product Backlog Items, which the Product Owner then in turn considers as candidates for the next Sprint.
Impediment		Anything preventing the Team from performing at their very best. Impediments are recorded on the Impediment Backlog and dealt with by the Scrum Master. Also called obstacles.
Impediment Backlog	IB	A list of current Impediment also showing their progress. The Impediment Backlog is owned by the Scrum Master.
Item		Short for Product Backlog Item.
Management		A secondary role in Scrum whose objective is to secure the success of the organization in which we find the Product Owner, Team and Scrum Master. The Manager typically assists in setting the Teams and removing impediments.
Milestone		A general Milestone in the Time-line for the Project, typically specifying that certain conditions must be met by this time, a special case is a Release Milestone.
PDSA cycle	PDSA	The Plan-Do-Study-Act cycle described by William Deming, the foundation for the work in Sprints with Sprint Review and Sprint Retrospectives to close the cycle with respect to the Product developed and the work processes.
Product		The result of the Project.
Product Backlog	PB	A sorted list of all requirements in the Project. General requirements are called Constraints and are kept outside the main Product Backlog. Typically the Product Backlog is sorted in a number of Releases of Sprints, indicating which Product Backlog Items are expected to be completed in which Sprints.
Product Backlog Item	PBI	A requirement or deliverable in the Product Backlog, typically formulated as a User Story with accompanying Acceptance Tests. Such an Item must as a minimum have a Business Value and an Estimate of cost associated to allow the Product Owner to order and prioritize the Product Backlog.
Product Burn-down		A graph showing how much work remains after each sprint. This allows forecasts of the expected finish time of the project to be made. It is typical also to include a graph showing the Team's Velocity in each Sprint, this again helps to forecast the next Sprint. Finally, it is also common to record the accumulation of Business Value, in order to allow an assessment of the prioritization done.
Product Owner	PO	A primary role in Scrum. This person assumes responsibility of Business Value and return of investment (ROI). The Product Owner owns the Product Backlog, anyone can add to it, but only the Product Owner moves items up and down in priority. The Product Owner is also the anchor person to turn to for explanation of Product Backlog Items.
Project		The complete undertaking consisting of Time-line, Constraints, Product Backlog, Product Owner, Team and Scrum Master
Release		A collection of Product Backlog Items that together make up a portion of the Product in some well-defined form and completeness that makes it useable to the User and Customer. A Release typically consists of several Sprints and can be associated with a Release Milestone in the Time-Line
Release Burn-down		The same as a Product Burn-down, but only with focus on a certain Release as opposed to the whole Product.
Release Milestone		A Milestone in the Time-Line associated with a Release
Release Sprint		The last Sprint before a Release, where certain different activities need to be performed (such as special deploy tests etc.). Some projects use this kind of Sprints. The Team must in these cases after each Sprint be prepared for that the Product Owner declares the next Sprint to be a Release Sprint.
Retrospective		Short for Sprint Retrospective

Scrum - a Bird's Eye View

Return on investment	ROI	A measure of the rate in which the Project generates value. Earlier value rather than later is of course preferable. This is a primary concern of the Product Owner.
Scrum		The name of the most popular Agile process; it takes its name from an event in the game of Rugby, where all players gather in a so-called Scrum to get a ball back into play.
Scrum Master	SM	A primary role in Scrum. This person assumes responsibility of the Scrum Process and for constant optimization. The Scrum Master ensures that everyone follows the process and does what he or she has committed to doing. He has a constant focus on optimization and removal of impediments; he owns the Impediment Backlog.
Scrum Team		Another name for the Entire Scrum Team, used by Ken Schwaber and Jeff Sutherland.
Selected Product Backlog		A collection of Product Backlog Items that the Team and the Product Owner have agreed to do in the upcoming Sprint. The selection is carried out during the Sprint Planning Meeting #1.
Sprint		A period of time in which the Team works on delivering a set of Product Backlog Items. A Sprint is sometimes also referred to as an iteration. A Sprint is normally between two and four weeks in time.
Sprint Backlog	SB	A list of Product Backlog Items selected for this Sprint together with the decomposed work (typically Tasks) that the Team has broken down the Product Backlog Items into. Each Product Backlog Item consists of one or more pieces of work. The Sprint Backlog shows the progress of these towards "Done".
Sprint Backlog Item	SBI	A piece of work belonging to a certain Product Backlog Item, a deliverable. These are pieces of work defined during Sprint Planning #2 (or during the Sprint). Normally work is broken down so that it only needs to be "In Progress" for one day, however, several Team members can work on it during that day.
Sprint Burn-down		A graph showing how much remaining work exists each day in the Sprint, this allows for forecasting if the Sprint is likely to end with all work completed, or if corrective action is called for.
Sprint Goal		A sentence describing the achievement of a Sprint; it is used to communicate to stakeholders and as a guideline if the Sprint plan has to be changed during the Sprint.
Sprint Planning		A term covering the two meetings Sprint Planning #1 and Sprint Planning #2
Sprint Planning #1		A planning meeting held at the very beginning of a Sprint. The Product Owner and the Team agree on a set of Product Backlog Items that the Team believes it can complete in this upcoming Sprint. Before entering the meeting the Product Owner has ordered the Product Backlog with the relevant Product Backlog Items at the top; he has also made sure that these are Estimated and Valued. The Team has beforehand assessed their capacity for the upcoming Sprint.
Sprint Planning #2		The second planning meeting at the beginning of a Sprint, where the Team analyses and designs how to deliver the selected Product Backlog Items. This produces a breakdown of work, typically series of Tasks. All this is placed on the Sprint Backlog which is the result of this meeting. At the end of this meeting, we have a commitment (forecast) from the Team and it is made public.
Sprint Retrospective		A meeting at the very end of the Sprint (after the Sprint Review), where the Team and Scrum Master analyze the completed Sprint and come up with ideas for improvement. They assess these and decide on which to do something about in the next Sprint. When this meeting is finished the Sprint is completed. This closes the PDSA cycle with respect to how work is performed.
Sprint Review		A meeting at the end of the Sprint, where the Team presents the results of the Sprint, the completed Product Backlog Items. The Product Owner and stakeholders (could be Manager, Customer, User and others) are present, the Product Owner approves (or disapproves) the results, new ideas or changes to the produced features are actively sought from Stakeholders. This closes the PDSA cycle with respect to how the Product should be like.

Story		A short form of User Story.
Story Point	SP	A unit-less measure of size of Product Backlog Items typically used in Scrum. It is used to estimate relative size of Product Backlog Items compared to each other. It is customary to use the Fibonacci numbers as a scale of Product Backlog Items called Stories (1, 2, 3, 5, 8 and 13) for larger ones (called Epics) it is customary to use 20, 40 and 100 as a scale.
Task		A very specific example of a Sprint Backlog Item.
Task-board		A typical implementation of the Sprint Backlog.
Team		A primary role in Scrum. These have assumed responsibility of transforming Product Backlog Items into finished pieces of final Product. The Team owns the Sprint Backlog, and keeps this updated at any time. The Team self-organizes during the Sprint and finds the right ways to implement the Product. The Team estimates Product Backlog Items, typically during an Estimation Meeting. The Team selects a collection of Product Backlog Items during Sprint Planning #1 together with the Product Owner and breaks these down into smaller pieces of work during Sprint Planning #2. After this a forecast and a commitment from the Team is presented. The Team is also called the Development Team.
Theme		A collection of Product Backlog Items that deal with the same general functional area of the Product Backlog.
Time boxing		The concept of allowing a specific time to an activity. When the time has elapsed, the activity is over, period.
Time-Line		The top-level view of time-based commitments in the whole Project. Normally the Time-Line consists of 1) "Milestones", where certain achievements must be reached, 2) "Deliveries In", where something is delivered to the Team and 3) "Deliveries Out", where the Team must deliver something to an outside party. Often Milestones take the form of a "Release Milestone", where the Product must exist in some well-defined form having reached a completeness making it useable to the User and Customer.
User		A secondary role in Scrum, who uses the Product, the result of the Project. User acceptance test are performed by Users or their proxies.
User Story		A way of describing what a Product Backlog Item is about, it often follows this template: "As such-and-such a user, I want to be able to do this-and-that in order to achieve jada-jada-jada." A User Story also has one or more Acceptance Tests connected to it, giving evidence as to when a User Story is Done.
Valuation		The process of assigning a numerical value indicating Business Value to a Product Backlog Item. The Product Owner is responsible for the Valuation, but may call on others to supply him with information (such as sales and marketing people or a customer forum).
Velocity		The Speed at which the Team delivers Product Backlog Items, typically measured in Story Points per Sprint.