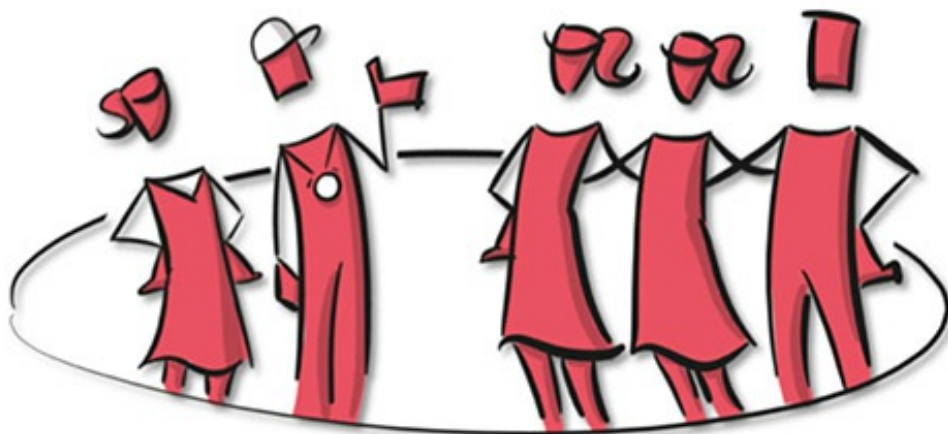
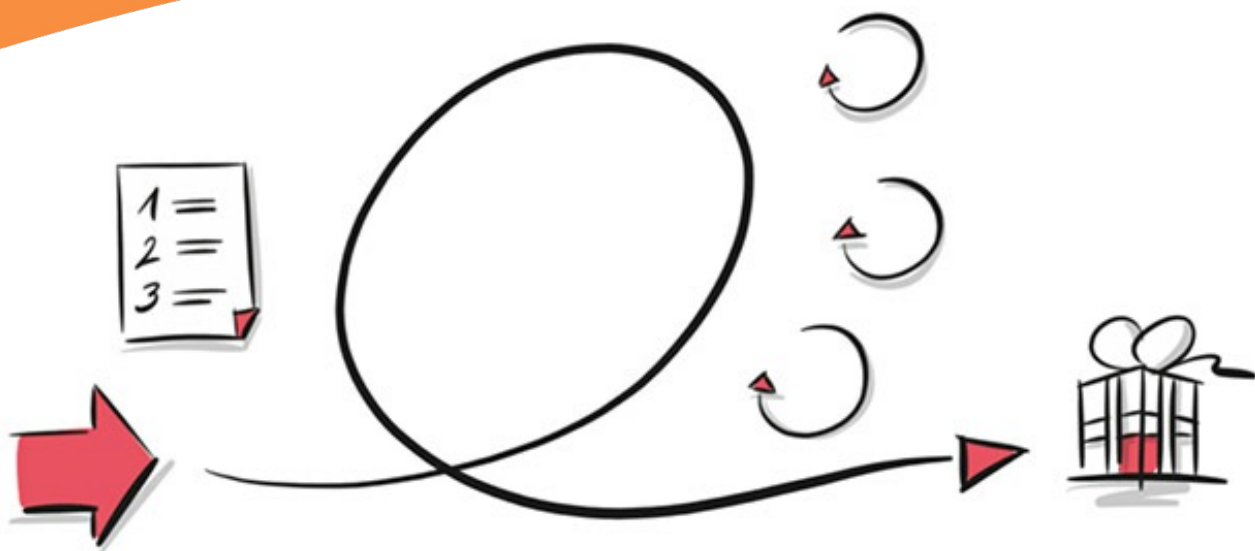


Scrum – an Introduction

A guide to the rules of the game and its history

Benjamin Sommer



BENJAMIN SOMMER

SCRUM – AN INTRODUCTION

A GUIDE TO THE RULES OF
THE GAME AND ITS HISTORY

Scrum – an Introduction: A guide to the rules of the game and its history

1st edition

© 2019 Benjamin Sommer & bookboon.com

ISBN 978-87-403-3090-8

CONTENTS

1	The author	6
2	What is Scrum?	8
3	Scrum and Agile	12
4	Scrum and Agile - a brief history lesson	14
5	Scrum theory	17
6	Scrum values	21
7	The Product Backlog – a Scrum artifact	22
8	The Scrum Team	25
8.1	The Product Owner	26
8.2	The Development Team	28
8.3	Development Team Size	29
8.4	The Scrum Master	30

www.sylvania.com

**We do not reinvent
the wheel we reinvent
light.**

Fascinating lighting offers an infinite spectrum of possibilities: Innovative technologies and new markets provide both opportunities and challenges. An environment in which your expertise is in high demand. Enjoy the supportive working atmosphere within our global group and benefit from international career paths. Implement sustainable ideas in close cooperation with other specialists and contribute to influencing our future. Come and join us in reinventing light every day.

Light is OSRAM

**OSRAM
SYLVANIA**

8.5	Combining roles?	32
8.6	What is the role of management in Scrum?	33
9	Scrum events	35
9.1	The Sprint	35
9.2	Sprint Planning	36
9.3	Sprint Backlog and Sprint Goal	39
9.4	Daily Scrum	43
9.5	Monitoring Sprint Progress	46
9.6	Sprint Review	48
9.7	Sprint Retrospective	51
10	Scrum artifacts – quick reference	55
10.1	Product Backlog	55
10.2	Sprint Backlog	56
10.3	Increment	56
11	References and further reading	57

1 THE AUTHOR

Benjamin Sommer is recognized worldwide as an expert in Agile and Scrum and he holds the two highest-level certifications in the field: CST (Certified Scrum Trainer) and CEC (Certified Enterprise Coach) by the Scrum Alliance. He has trained over 1000 students and coached a variety of organizations; small, large, private, public and government throughout Europe and USA. His experience comes from management consulting and cofounding and building companies in the role as cofounder, angel investor and board member. His focus is on helping organizations achieve business agility, make product teams go faster and deliver more value. Benjamin has more than 20 years' experience as business developer, management consultant, entrepreneur and investor. He is an investor, business development consultant, speaker, trainer/facilitator and occasional writer.

Benjamin most recently received a Certificate of Business Excellence from University of California Berkeley, Haas School of Business. He holds an MBA from the University of Agder and Executive education from BI, Norwegian School of Business.

Benjamin prides himself on sharing his knowledge and experience so feel free to contact him if you have questions, comments or need assistance. Benjamin can be contacted via his LinkedIn profile <https://www.linkedin.com/profile> or through www.leanventure.com.

Online business card: www.about.me/benjaminsommer

Request a Certified Scrum Master, Certified Scrum Product Owner or a custom agile/Scrum/business model innovation event in your city or in your organization by contacting the author.



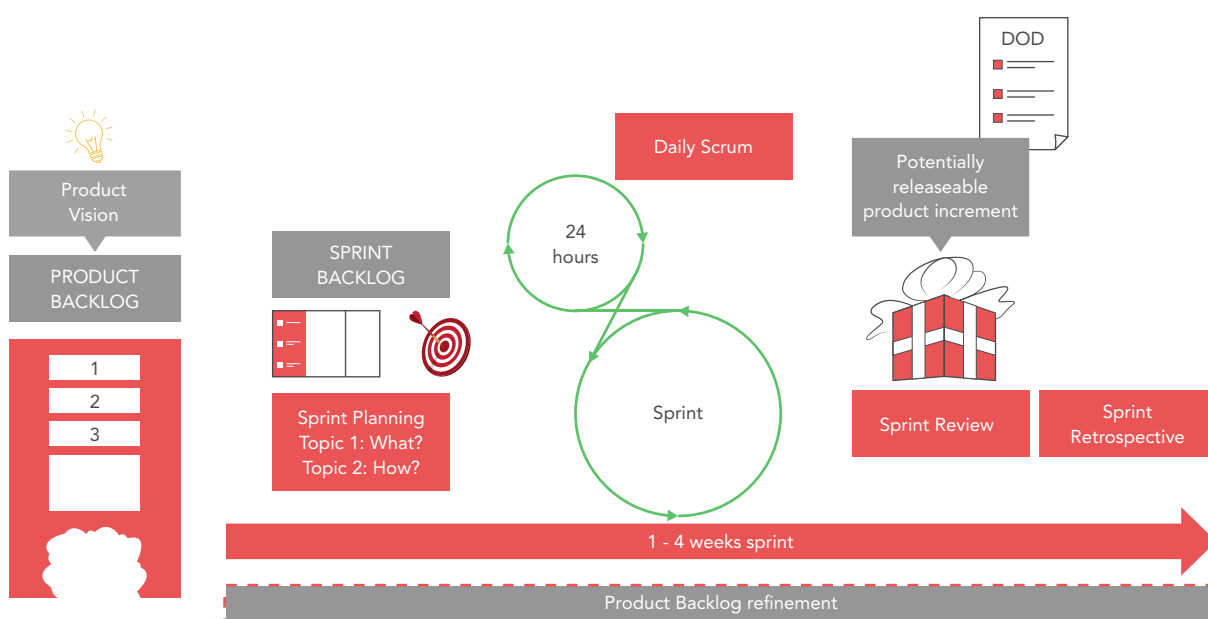


E-mail: benjamin@leanventure.com Phone: +47 95 24 28 95

Skype: benjamin_sommer_ Twitter: @benjaminsommer

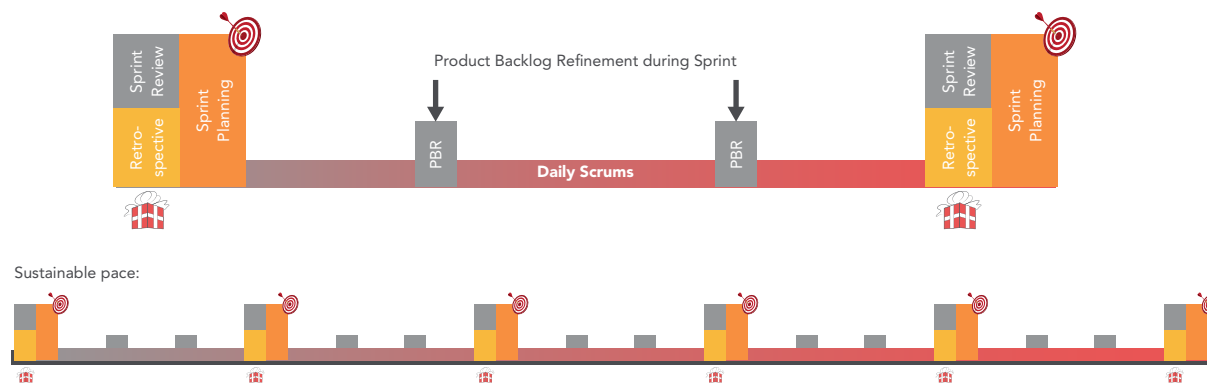
2 WHAT IS SCRUM?

Scrum is a framework for developing, delivering, and sustaining complex products. The Scrum framework consists of roles, events, artifacts, and rules that govern the interaction of the roles, events and artifacts. Scrum was initially developed for managing and developing products. Starting in the early 1990s, Scrum has been used extensively, worldwide to develop software, hardware, embedded software, autonomous vehicles, schools, government, marketing, managing the operation of organizations and also to manage our daily lives. Scrum is the most widely used agile framework in the world today. This eBook will explain the simple yet, challenging nature of the Scrum framework, its relation to agile and outline their history.



The essence of Scrum is the Scrum team that works in an incremental and iterative fashion to maximize learning and integrate these learnings into the product/service and way of working. Scrum is **iterative** and **incremental**. Scrum structures development in a cycle of work (iteration) called Sprints. Each Sprint and its events are formal opportunities for the Scrum Team to inspect and adapt. Since development inevitably involves learning, innovation, and surprises, Scrum emphasizes taking a short step of development, **inspecting** both the resulting product and the efficacy of current practices, and then **adapting** the product goals and process practices, see figure 1. *Repeat forever.*

The Sprints in Scrum are one to four weeks, with the most common being two weeks. Sprints take place one after the other without pause. The Sprints are **timeboxed** – they end on a specific date whether the work has been completed or not, and are never shortened or extended. Usually Scrum Teams choose one Sprint length and use it for all their Sprints until they improve and can use a shorter cycle.



Each Sprint starts with a Sprint Planning where a cross-functional, self-organizing Scrum team selects items (product/customer requirements) from a prioritized list called the Product Backlog. The Scrum Team agrees collectively on a goal of what they believe they can deliver by the end of the Sprint, something tangible that will be truly “done”. During the Sprint, no new items may be added; Scrum embraces change for the next Sprint, but the current short Sprint is meant to focus on a small, clear and stable Sprint goal.

Why is it called Scrum?

The name Scrum comes from rugby and was first mentioned in a paper published in the Harvard Business Review titled “The New New Product Development Game” (H. Takeuchi and I. Nonaka, 1986). Nonaka & Takeuchi used the analogy of rugby to describe how teams that successfully innovated was cross-functional, moving as one unit up the field and passing the ball within the team with a clear and common purpose. The term Scrum was later used by Jeff Sutherland and Ken Schwaber to describe their approach to software development.

Every day the Development Team gathers briefly to inspect its progress towards the Sprint Goal, and adjust the next steps needed to complete the work remaining. At the end of the Sprint, the Scrum Team reviews the Sprint with stakeholders, and demonstrates what it has built. People obtain feedback that can be incorporated in the next Sprint. Scrum emphasizes working product at the end of the Sprint that is really “done”; in the case of software, this means a system that is integrated, fully tested, end-user documented, and potentially shippable.

Key success criteria for iterative and incremental development**Every increment:**

- Is testable / usable
- Adds value
- Is high quality
- Is ready to be released

At the end of each iteration:

- Progress is reviewed and plans adjusted
- The product is returned to a stable state
- The product can be released as is
- A go/no-go decision on further development can be made

Scrum is *lightweight* and *simple to understand*, yet it is *difficult to master*. It is lightweight in the sense that the framework is minimalistic and intentionally incomplete. Minimalistic in the sense that it is the bare minimum required to ensure feedback and continuous learning. Intentionally incomplete in the sense it is a framework within which you can employ various processes and techniques. Scrum should be combined with other practices and tools to form a complete process. Scrum makes clear the relative efficacy of your product management and work techniques so that you can continuously improve the product, the Scrum team, and the working environment. Scrum leaves it to the Scrum team to fill the gaps and decide on practices, tools and techniques. This autonomy will give the team ownership of its process, a feeling of responsibility and lead to continuous improvement when nurtured carefully.

The roles, events, artifacts, and the rules that govern them are simple to understand, yet difficult to master. The essence of Scrum is a small cross-functional, self-organizing team of people. The individual team is highly flexible and adaptive. The team should also be stable and long-lived. Scrum requires change and most likely it won't work in your organization - not the way your organization is working now.

Scrum significantly impacts how work is organized in two ways. First, having the team as main organizational unit changes structures, roles, decisions processes and behaviors in the organization. The Scrum team have all the capabilities and knowledge its need to make decisions and should not be reliant on anyone outside of the team. The Scrum team choose themselves how to best accomplish the goals and objectives presented to them by the organization. Secondly, Scrum manage work as products rather than projects, again changing structures, decisions, and behaviors in product development. For Scrum to be used successfully in your organization both of these impacts must be understood.

Scrum as defined exist only in its entirety. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage. The skillful use of Scrum requires discipline and dedication.

To sum up, the Scrum Team's members (the Product Owner, the Development Team, and the Scrum Master) collaborate to create a series of Product Increments, during short time-boxed intervals called Sprints. Each increment meets the Product Owner's acceptance criteria and the team's shared "Definition of Done". They work from a Product Backlog. In each Sprint, they begin with Sprint Planning to produce the Sprint Backlog, a plan for the Sprint. They self-organize to do the Development, using Daily Scrum meetings to coordinate and to ensure that they produce the best possible Product Increment. They perform Product Backlog Refinement to prepare for the next Sprint's planning meeting. They end the Sprint with the Sprint Review and Sprint Retrospective, reviewing the product and their process.

What is an artefact?

An artifact is usually simple object (such as a tool or ornament) showing human workmanship or modification as distinguished from a natural object (Marriam-Webster).

Scrum have three required artefacts; Product Backlog, Sprint Backlog and Increment, each of which represent work or value to provide transparency and opportunities for inspection and adaption.

The **Product Backlog** is a list of everything that is known to be needed in the product. It is owned and managed by the Product Owner.

The **Sprint Backlog** is a plan for the current Sprint on how to deliver the next product Increment. It is owned and managed by the Development Team.

The **Increment** is the product your organization is building and maintaining. The Increment is the sum of all Product Backlog items completed during a Sprint and the value of all previous Sprints.

3 SCRUM AND AGILE

Scrum is a framework that has been used to manage work on complex products since the early 1990s. Scrum was initially developed for managing and developing products. Starting in the early 1990s, Scrum has been used extensively, worldwide to develop software, hardware, embedded software, autonomous vehicles, schools, government, marketing, managing the operation of organizations and also to manage our daily lives.

Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. Agile is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it.

Scrum is based on the agile values and principles and is as such an agile framework. One way to think of the relationship between Scrum and agile is to think of agile as a mindset and Scrum as framework within the agile mindset. Scrum and agile share the focus on the people doing the work and how they work together. Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context.

In 2001 a group of seventeen men got together in Snowbird, Utah to discuss the growing field of what was then known as lightweight methods. They decide to use the term agile as the label because that word represented the adaptiveness and response to change which was so important to their approach. They also wrote the Manifesto for Agile Software Development, setting out the values and principles of agile. See the next pages for the Agile Manifesto and the 12 agile principles.



The twelve principles behind the agile manifesto

We follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

4 SCRUM AND AGILE - A BRIEF HISTORY LESSON

Agile has a long history and projects using an iterative and incremental approach can be traced back to the 1950s and 1960s. Agile has also been greatly influenced by the Lean movement from Honda and Toyota in Japan. One of the very first mentions of Scrum was in the 1986 Harvard Business Review article “The New New Product Development Game” by Ikujiro Nonaka and Hirotaka Takeuchi.

Nonaka & Takeuchi described a new approach to commercial product development that would increase speed and flexibility, based on case studies from manufacturing firms in the automotive, photocopier and printer industries. They called this the holistic or rugby approach, as the whole process is performed by one cross-functional team across multiple overlapping phases, in which the team “tries to go the distance as a unit, passing the ball back and forth”. (In rugby football, a scrum is used to restart play, as the forwards of each team interlock with their heads down and attempt to gain possession of the ball).

In the early 1990s, Ken Schwaber used what would become Scrum at his company, Advanced Development Methods; while Jeff Sutherland, John Scumniotales (the first ever Scrum Master) and Jeff McKenna developed a similar approach at Easel Corporation, referring to it using the single word Scrum. The primary driver for beginning the first Scrum at Easel Corporation was an absolute commitment to a date, where failure would break the company. Easel Corporation had to guaranteed delivery of an innovative product to the market that would achieve rapid adoption.

Jim Coplien in a 1994 article called “Borland Software Craftsmanship: A New Look at Process, Quality and Productivity” described his observations of the “hyperproductive” Borland Quattro Pro team, noting their reliance on almost daily meetings: “the project was made more of meetings than anything else”. This article also strongly influenced Scrum. The Nonaka & Takeuchi and the Coplien article combined triggered the first Daily Scrum meetings and a monthly Sprint cycle.

In 1995, Sutherland and Schwaber jointly presented a paper describing the scrum framework at the Business Object Design and Implementation Workshop held as part of Object-Oriented Programming, Systems, Languages & Applications '95 (OOPSLA '95) in Austin, Texas. The notion of the Sprint as iteration was also introduced in this paper.

In 1997 Ken Schwaber describes the “daily scrum” (which does not appear in his earlier writings, such as the 1995 article “SCRUM Development Process”), this is later recast in pattern form by Mike Beedle. The Standup meeting pattern was first described by Jim Coplien in 1993.

The burndown chart was first described by Ken Schwaber in 2000. He engineered it while working at Fidelity Investments in an attempt to provide Scrum teams with a simple tool kit. Burndown charts are not part of core Scrum but a good practice often used by Scrum teams.

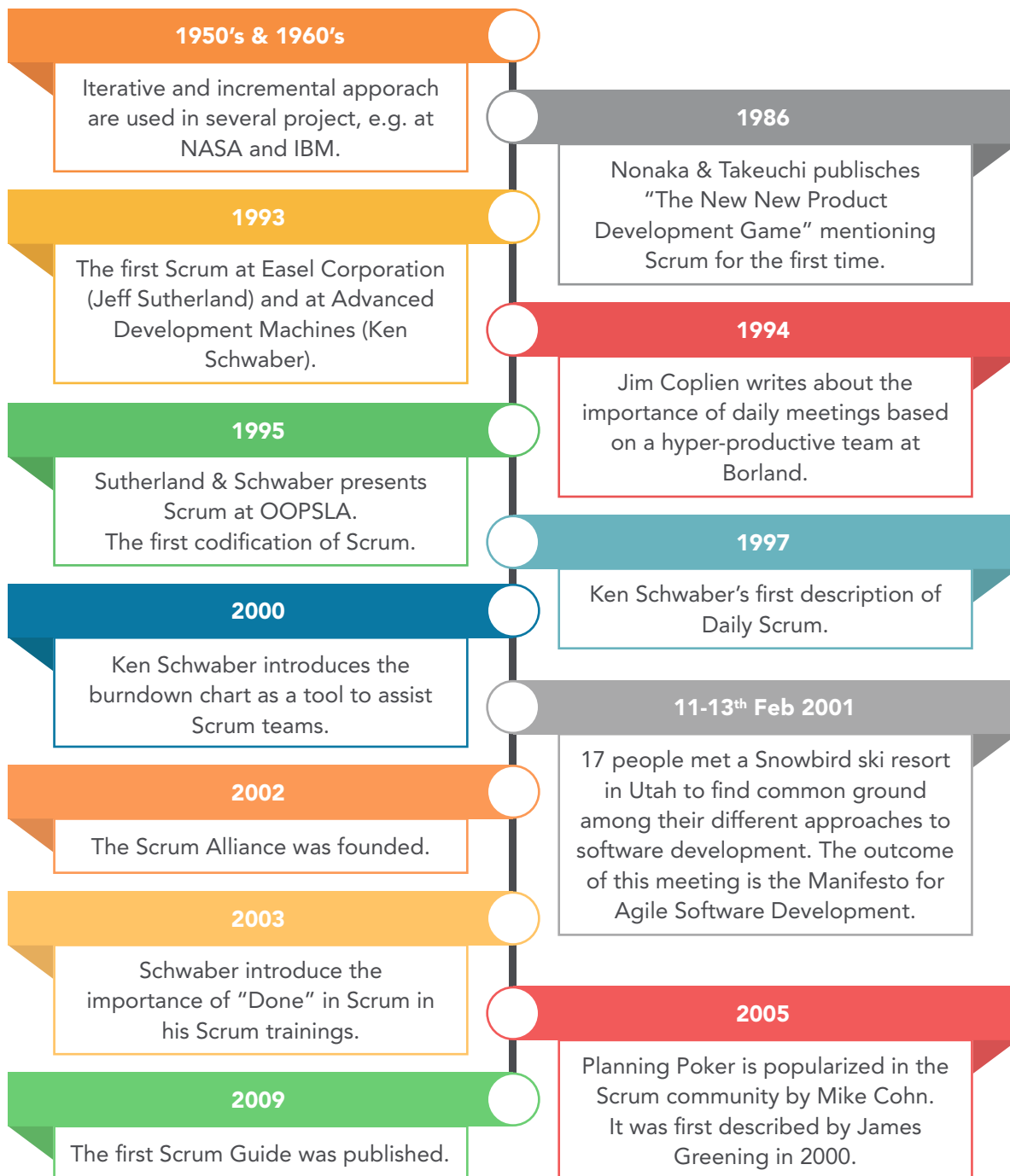
February 11-13 2001: 17 people who develop software and help others do it met at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah to find common ground among their different approaches to software development. The outcome of this meeting is the Manifesto for Agile Software Development. Several members of that discussion went on to found the Agile Alliance.

In 2001, Schwaber worked with Mike Beedle to describe the framework in the book, Agile Software Development with Scrum. As the Scrum community started growing, it was decided to create a platform to bring them together, which in turn led to the birth of the Scrum Alliance (SA) and Certified ScrumMaster (CSM) certification in 2002. Scrum Alliance remains the largest, most established and influential professional membership and certification organization in the agile community and has certified more than 750,000 practitioners worldwide.

Mary Poppendieck’s article, “Lean Programming”, published in 2001, draws attention to the structural parallels between Agile and the ideas known as Lean or the “Toyota Production System”.

From 1995 to 2009, Schwaber and Sutherland collaborated to combine their material with their experience and evolving good practice, to develop what is today known as Scrum. Several others had notable contributions around product backlog refinement, Definition of Done, daily meeting, task board, burnup chart and retrospectives.

In 2009 the Scrum Guide, a public document officially defining Scrum was first published. The Scrum Guide was written by Ken Schwaber and Jeff Sutherland and they still maintain the official Scrum Guide. It has been revised 5 times, with the current version being November 2017.



5 SCRUM THEORY

Scrum is specifically designed to manage risk and optimize predictability in complex adaptive systems. The iterative and incremental approach Scrum applies makes it possible to quickly adapt to changes in the environment, unexpected feedback and incorporate our learning both about our product or service and our process to produce these products and services.

A complex system is a system composed of many components which interact with each other. The behavior of complex systems is intrinsically difficult to model due to the dependencies, competitions, relationships, or other types of interactions between their parts or between a given system and its environment. Complex adaptive systems are special cases of complex systems that are adaptive in that they have the capacity to change and learn from experience. Examples of complex adaptive systems include the stock market, manufacturing businesses and any human social group-based endeavor such as business or public organizations.



Discover the truth at www.deloitte.ca/careers

Deloitte.

© Deloitte & Touche LLP and affiliated entities.

Complex adaptive systems are dynamic networks of interactions, and their relationships are not the aggregations of the individual static entities, i.e., the behavior of the ensemble is not predicted by the behavior of the components. They are adaptive in that the individual and collective behavior mutate and self-organize corresponding to the change-initiating micro-event or collection of events.

In an organizational context where we have groups of people involved and they rely on interactions with each other to accomplish their goal we are dealing with a complex adaptive system. As technology, market, and environmental complexities and their interactions have rapidly increased, Scrum's utility in dealing with complexity is proven daily.

Since we, in a complex adaptive system, are unable to perfectly predict the detailed task necessary to achieve a specific outcome nor the specific individual deliverables to achieve a specific outcome Scrum have built in feedback loops with adaptive planning cycles to incorporate our learning as we progress and improve.

Scrum is founded on empirical process control theory. With empirical process control we gain knowledge from experience, enabling us to make decisions based on what is known. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

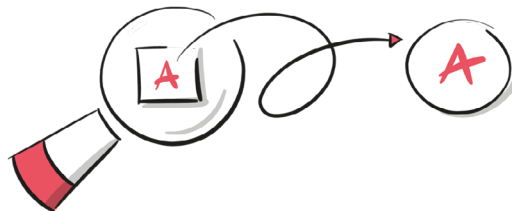
Transparency: Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen. For example, a common language referring to the process must be shared by all participants; and those performing the work and those inspecting the resulting increment must share a common definition of what it means to be “Done”. Known in Scrum as the Definition of Done. The three Scrum artifacts, Product Backlog, Sprint Backlog and Increment helps us achieve transparency.

Inspection: Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

Adaptation: If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation;

1. Sprint Planning
2. Daily Scrum
3. Sprint Review
4. Sprint Retrospective



The Scrum events are described in more detail in chapter 9.

Scrum relies on transparency. Decisions to optimize value and control risk are made based on the perceived state of the artifacts (see page 11 or chapter 10 for more information about Scrum artifacts). To the extent that transparency is complete, these decisions have a sound basis. To the extent that the artifacts are incompletely transparent, these decisions can be flawed, value may diminish and risk may increase.

The Scrum Master must work with the Product Owner, Development Team, and other involved parties to understand if the artifacts are completely transparent. There are practices for coping with incomplete transparency; the Scrum Master must help everyone apply the most appropriate practices in the absence of complete transparency. A Scrum Master can detect incomplete transparency by inspecting the artifacts, sensing patterns, listening closely to what is being said, and detecting differences between expected and real results.

The Scrum Master's job is to work with the Scrum Team and the organization to increase the transparency of the artifacts. This work usually involves learning, convincing, and change. Transparency doesn't occur overnight, but is a path.

Used correctly Scrum will over time creating a learning, resilient organization that can move fast and with great maneuverability.

"The only sustainable competitive advantage is an organization's ability to learn faster than the competition."

– Peter Senge

The Product Increment

The product or service that your organization creates is an artifact in Scrum referred to as the Increment. The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. Hence, the Product Increment includes the functionality of all previous Product Increments and is fully tested so that all completed Product Backlog items continue to work together. The increment is a step toward a vision or goal. At the end of a Sprint, the new Increment must be “Done,” which means it must be in useable condition and meet the Scrum Team’s definition of “Done.”

Definition of Done

The definition of “Done” is different for every Scrum Team, however members must have a shared understanding of what it means for work to be complete. This ensures transparency and aligns the Development team on quality. The Definition of Done is a standard for any work done on a product or system, a shared quality standard the Development team must meet before work can be called complete. Definition of Done should include the quality standard that are so important every Product Backlog Item must comply with them.

The Definition of Done must always include the notion that the Product Increment is of high enough quality to be releasable. The Product Owner could choose to release it immediately. As a Development team matures, the Definition of Done will expand and become more stringent. If multiple Development Teams working on the system or product release, they must have a shared definition of “Done” to ensure transparency of the Increment across teams.

The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team’s current definition of “Done.”

6 SCRUM VALUES

The essence of Scrum is a cross-functional, self-organizing team of motivated individuals. For a group of individuals to form a team, where they put the team first and themselves second, we need a shared set of values and a strong sense of purpose. Through the use of teamwork and continuous improvement, Scrum both creates these values and relies on them. The values are Focus, Courage, Openness, Commitment, and Respect. These values are fundamental to build trust among the team members and between the Scrum team and their surroundings.

FOCUS: Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.



OPENNESS: As we work together, we practice expressing how we're doing, and what's in our way. We learn that it is good to express concerns, so that they can be addressed.

RESPECT: As we work together, sharing successes and failures, we come to respect each other, and to help each other become worthy of respect.

COMMITMENT: Because we have great control over our own destiny, we become more committed to success.

COURAGE: Because we are not alone, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges.

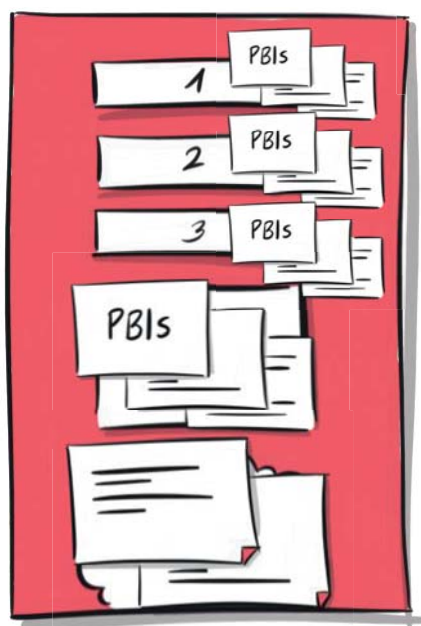
When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum events, roles and artifacts.

Successful use of Scrum depends on people becoming more proficient in living these five values. People personally commit to achieving the goals of the Scrum Team. The Scrum Team members have courage to do the right thing and work on tough problems. Everyone focuses on the work of the Sprint and the goals of the Scrum Team. The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work. Scrum Team members respect each other to be capable, independent people.

7 THE PRODUCT BACKLOG – A SCRUM ARTIFACT

Scrum's artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

The Product Backlog is one of the three essential artifacts in Scrum. The Product Backlog is an ordered list of everything known to be needed in the product. The Product Owner is responsible for ordering the Product Backlog to maximize the value of the product and the work the Development team performs. The Product Backlog provides transparency for the entire Scrum team, stakeholders and wider organization about the planned future developments of the product Increment.



The Product Backlog is the single source from which all requirements flow. This means that all work the Development Team does comes from the Product Backlog. Every feature idea, enhancement, bug fix, technical upgrades, every bit of work they do, is derived from an item in the Product Backlog (referred to as Product Backlog Items or PBIs). Each item on the Product Backlog should have a description, an estimate, an order and acceptance criteria. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering. Even so, a good Product Owner should have help in producing and keeping the Product Backlog current. Product Backlog items may originate from the Product Owner, from Development team members, or from other stakeholders.

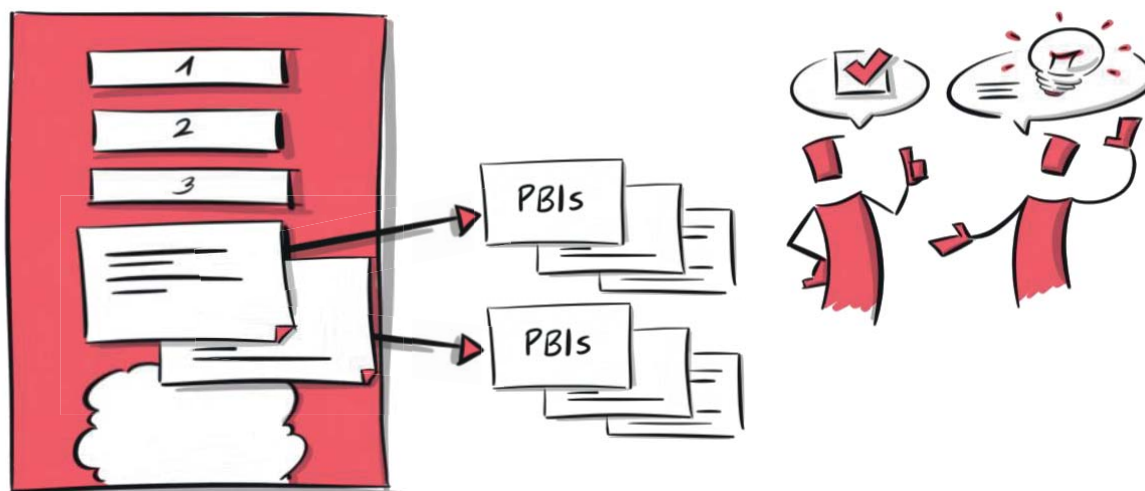
The Product Backlog is a living artifact and will evolve and emerge through the lifecycle of the product. It may begin as a large list or a short one. It may be vague or rather detailed. The Product Backlog typically begins short and vague and becomes a larger and more exhaustive list as the marketplace provides feedback. The Product Backlog and its earliest development contents should be derived from a shared, engaging and clear product vision.

The Product Backlog is never complete. It is dynamic, with constant changes to identify what the product needs to be appropriate, competitive, useful and valuable. Changes in business requirements, market conditions, or technology cause changes in the Product Backlog. If a product exists, its Product Backlog also exists.

If multiple Scrum Teams work together on the same product, they will work from the same Product Backlog.

Empirical product development – ongoing refinement of the Product Backlog

Since the Product Backlog is never complete, dynamic and constantly changes, keeping the Product Backlog healthy is an ongoing activity. A healthy Product Backlog is ordered, detailed appropriately, estimated and emerges as we learn more. A Product Backlog that is detailed appropriately will have items on top of the Product Backlog that are ready to take into a Sprint. Typically, there should be enough Product Backlog items ready about the size of two to three Sprints of work. This ensure that we have some flexibility. These items are relatively small, well understood and have acceptance criteria. A Development Team should be able to pull 5-10 PBIs into a Sprint and complete them within the Sprint time-box. Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail.



Further down in the Product Backlog will be items that quite often are large and general in nature. The activity of clarifying, better define, split into smaller items and estimating are all part of the Product Backlog Refinement activity. The Development Team, the people who will perform the work, is responsible for all estimates.

As part of the feedback we get on your product and changes in the marketplace keeping the Product Backlog healthy also includes removing or demoting items that no longer seem important. A great Product Owner will treat the Product Backlog with suspicion, challenging assumptions, importance and value of features and enhancements.

Product Backlog Refinement is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team. However, Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

One key benefit of the Product Backlog Refinement activity is to prepare for upcoming Sprints. Product Backlog items that are deemed "Ready" for a Sprint should be clear, feasible to complete within the Sprint time-box, have acceptance criteria and each item entering the Sprint should represent an increment of "business value".

8 THE SCRUM TEAM

Scrum is a proven framework for product success in organizations. Becoming a truly agile organization requires a significant change in organizational culture. People play a critical role in ensuring the success of products by meeting aggressive deadlines and stepping up to complex demands. The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master.


Product Owner; the product value evangelist - holds the vision for the product and is responsible for maximizing the value of the product

Development Team; the collaborative friendly collective - builds the product and determines how to deliver value


Scrum Master; creator of transparency, focused on others growth and continuous improvement - mentors, facilitates and coaches the team to the best use of Scrum resulting in transparency and continuous improvement.

SIMPLY CLEVER

ŠKODA



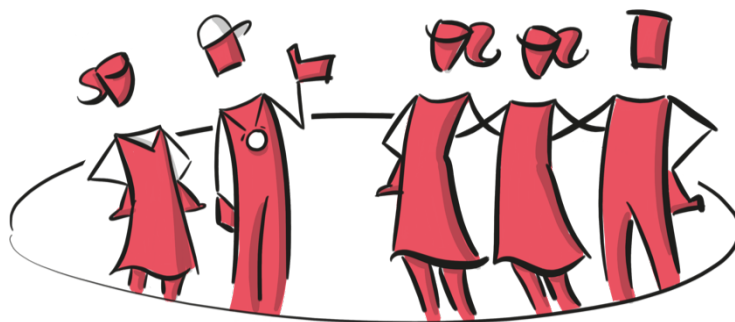
We will turn your CV into an opportunity of a lifetime



Do you like cars? Would you like to be a part of a successful brand? We will appreciate and reward both your enthusiasm and talent. Send us your CV. You will be surprised where it can take you.

Send us your CV on
www.employerforlife.com





Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback, while reducing risk and optimizing predictability. Incremental deliveries of “Done” product ensure that a potentially useful version of the working product is always available.

Let’s look at how each of these roles contributes to a product’s success, the challenges encountered along the way, and some of the personality traits that drive effective solutions.

8.1 THE PRODUCT OWNER

The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. The Product Owner owns the product vision, roadmap, release plan and long-term product strategy.

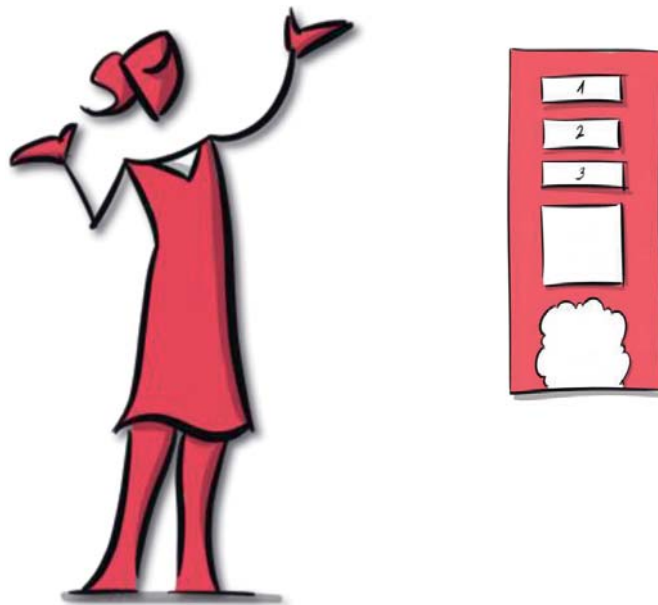
The Product Owner is the single individual who is responsible for drawing out the most valuable possible product by the desired date. This is done by managing the flow of work into the team, selecting and refining items from the Product Backlog. The Product Owner maintains the Product Backlog and ensures that everyone knows what is on it and what the priorities are. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may be supported by other individuals but must be a single person. The Product Owner may represent the desires of a wide variety of stakeholders in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner.

Certainly, the Product Owner is not solely responsible for everything. The whole Scrum Team is responsible for being as productive as possible, for improving their practices, for asking the right questions, for helping the Product Owner, and so on. The Development Team is responsible for determining how much work will be taken on in a Sprint, and for producing a usable Product Increment every Sprint. The Development team will also assist the Product Owner in working with the Product Backlog (known as Product Backlog refinement), however, the Product Owner remains accountable.

The Product Owner, in Scrum, is in a unique position. The Product Owner is typically the individual closest to the “business side”. The Product Owner is often charged by the organization to “get this product out”, and is the person who is expected to do the best possible job of satisfying all the stakeholders. The Product Owner does this by managing the Product Backlog, and by ensuring that the Product Backlog, and progress against it, is kept visible.



The Product Owner, by choosing what the Development Team should do next and what to defer, makes the scope versus schedule decisions to lead to the best possible product. For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and ordering of the Product Backlog. No one can force the Development Team to work from a different set of requirements.

On a practical level, a Product Owner is responsible for defining the product vision, setting priorities to deliver the highest value and determining what product deliverables are important to a wide variety of stakeholders.

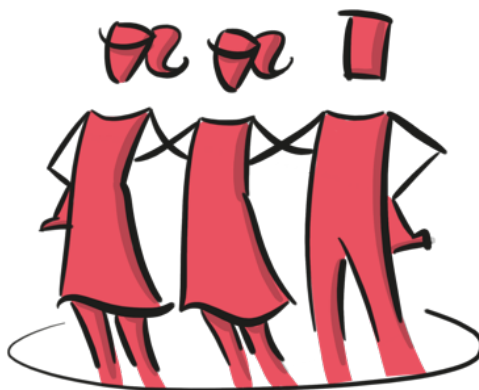
But in today's fast-paced, highly competitive world, a Product Owner must also be a visionary. Fluctuating needs and ongoing feedback can change the direction of a product. For this reason, a product owner must also be a good communicator. Not only does this involve the product owner explaining changing priorities and their varying impacts to different product teams, it also means listening to stakeholders and carefully managing their expectations.

Not everyone is built for the role of Product Owner. A Product Owner must be allowed to make critical decisions, no matter how unpopular. Courage is essential. After all, creating a vision for a product and determining a direction forward requires making choices that may encounter dissent, and persevering in the overall interest of maximizing value—both of the product, and for the customer.

Product Owners must also be highly self-disciplined. It can be tempting to try to control the work of others. Experienced product owners know not to try to manage the Scrum team's activities.

8.2 THE DEVELOPMENT TEAM

The development team is a self-organizing collective — a cross-functional team that includes all the roles and skills required to complete a Product Backlog item. From architects and testers to developers and designers, these self-organizing teams are accountable for delivering a potentially releasable Increment of “Done” product at the end of each Sprint. The Development team delivers chunks of working product in frequent, iterative increments. A “Done” increment is required at every Sprint Review. Only members of the Development Team create the Increment.



Self-organizing in the context of agile means the Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality.

Because of the self-organizing nature of a Scrum team, members must be willing to share ownership of the work. Desirable traits include an eagerness to collaborate, technical excellence, attention to detail and adaptability. Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

The Product Owner and the priorities he or she sets for each Sprint are what determine the specific features that a Development team work on at any given time. For example, Development Team members can take a feature from an ordered Product Backlog and collaboratively decide how to develop the solution. This level of autonomy encourages strong bonds among team members and helps to create a positive working environment and creative solutions.

By placing knowledgeable team members close to a complex product and its customers, organizations can react quickly to changing circumstances. Self-organizing development teams can deliver enormous value in a month or less by learning quickly without feeling overworked for a rapid return on investment.

8.3 DEVELOPMENT TEAM SIZE

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. Fewer than three Development Team members decrease interaction and results in smaller productivity gains. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Having more than nine members requires too much coordination. Large Development Teams generate too much complexity for an empirical process to be useful. Only the members of the Development team are included in this count.

8.4 THE SCRUM MASTER

The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values. Scrum relies on transparency (chapter 5 Scrum theory) and decisions to optimize for value and control risk should be made based upon the state of the artifacts (Product Backlog, Sprint Backlog and product Increment).



The Scrum Master works with the Scrum team and the organization to ensure the artifacts are transparent enabling continuous learning and improvement based on shared insights and facts. In the absence of complete transparency, it's the Scrum Masters responsibility to help the Development team, the Product Owner and the organization as a whole to apply the most appropriate practices to improve transparency. In this way the Scrum Master helps the Scrum team and the organization to perform at its highest level. However, it is extremely important to understand that the Scrum Master do not create transparency by assuming the responsibilities of the Product Owner and/or the Development team themselves. Rather, the Scrum Master works through the Development team, the Product Owner and the entire organization displaying an agile mindset and living the Scrum values.

The ScrumMaster must have a good understanding of the Scrum framework and the ability to train others in its subtleties. The Scrum Master teaches and guides the Scrum team on how to use Scrum to deliver a valuable product. An expert on how Scrum works, he or she must help everyone employ the Scrum framework and facilitate proper application of Scrum. The Scrum Master should support all team members in using Scrum effectively. Using Scrum effectively will result in transparency.

The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team. This way the

Development team members can focus on their work without any distractions. The Scrum Master role also involves meeting the Product Owner and Development team members where they are, with their individual strengths, and helping them deploy those strengths in achieving a shared objective. Therefore, this role requires someone who is willing to have tough conversations and to abandon his or her own agenda for the sake of product success.

A great Scrum Master will seamlessly adapt his role to be a facilitator, mentor, teacher, coach or change agent based on the context and situation. The Scrum Master acts as a coach for the Scrum Team, helping the team to work together and learn the Scrum framework. He may facilitate meetings, and helps keep the Scrum Team on track, productive, and growing in ability. The Scrum Master helps everyone improve to make the Scrum Team more productive and valuable.

The Scrum Master is a servant-leader for the Scrum Team. A servant-leader puts the needs of others first and helps people develop and perform as highly as possible. Servant-leaders are skilled communicators, compassionate collaborators, use foresight, and are systems thinkers. A true servant leader will measure his own success on the growth of the people and team they serve.

The Scrum Master works with the Product Owner to help the Product Owner understand how to create and maintain the Product Backlog. The Scrum Master serves the Product Owner in several ways, including ensuring that goals, scope, and product domain are understood by everyone on the Scrum Team as well as possible. A great Product Owner will also have some facilitation skills but the Scrum Master might help and also assist in finding techniques for effective Product Backlog management and how to arrange the Product Backlog to maximize value.

The Scrum Master will also coach the Development Team in self-organization and cross-functionality. He works with the Development Team to find and implement the technical practices that will allow them to get to done at the end of each Sprint, helping the Development Team to create high-value products by facilitating and coaching.

The Scrum Master will work with the whole Scrum Team to evolve the Definition of Done.

In any team and organization there will be impediments to the Scrum teams progress. These impediments can be internal to the Scrum team, i.e. the Product Owner is not available or external to the Scrum team, such as individual bonus schemes, part-time team members due to focus on resource utilization etc. It's the responsibility of the Scrum Master to ensure impediments like these are removed. However, the Scrum team is self-organizing and impediments and issues should be removed by the Scrum team if possible.

The Scrum Master should also focus on the organization as a whole and have a responsibility in coaching, mentoring and teaching the organization towards Scrum. A Scrum Master should lead and coach the organization in its Scrum adoption, help employees and stakeholders understand and enact Scrum and empirical product development and cause change that increases the productivity of the Scrum Team.

If there are several Scrum Masters in an organization a resourceful Scrum Master would work with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

Visual management: A Lean approach to transparency

Visual management or visual controls is a central tool in Lean and is one of the 14 principles of the Toyota Way. Visual Management is the practice of using information visualization techniques to manage work. Many visual management ideas come from traditional Lean thinking and Toyota, but these techniques are also very popular within the Agile Software Development community. In 2000 Alistair Cockburn, one of the signatories of the agile manifesto, coined the term "information radiators". An Information radiator is a display posted in a place where people can see it as they work or walk by. It shows readers information they care about without having to ask anyone a question. This means more communication with fewer interruptions.

A good information radiator;

- Is large and easily visible to the casual, interested observer
- Is understood at a glance
- Changes periodically, so that it is worth visiting
- Is easily kept up to date

Typically, it is on paper, posted in the team room or in the hallway.

Scrum relies heavily on making information transparent to have an effective inspect and adapt feedback loop. Visual management is a highly recommended practice successful Scrum teams embrace.

8.5 COMBINING ROLES?

Combining roles in Scrum is very difficult and will lead to frustration for the Scrum team. It will deeply impact the benefits of Scrum most likely also directly affect business value. The three Scrum roles are all full-time roles and combining any two will diminish either one, or in my experience, or both of the roles being combined. Let's quickly look at the different possible combinations.

Product Owner and Scrum Master: The Product Owner is responsible for driving the Development team through the work in the Product Backlog. The Scrum Master should facilitate the dialogue between the Product Owner and the Development team. If there are some disagreement of challenges in the relationship between the Product Owner and the Development team the Scrum Master is not there to balance the interaction. Also, a Product Owner/ Scrum Master combination will diminish both roles and neither would be performed in a good way.

Product Owner and Development team member: A Product Owner will meet with customers and stakeholders to elicit needs, requirements and challenges with the current product. The Product Owner needs to build and maintain a healthy Product Backlog to maximize value. As member of the Development team your fellow team members expect you to be working as part of the Development team full time to realize the Sprint Goal. In addition, the faster the Development team goes, the harder it will be for the Product Owner to keep up. A Product Owner has to filter too much noise to be able to focus effectively on the work required of a Development team member. There is also a possibility for misunderstandings and confusion as it is difficult for the other member of the Development team to know if the person is talking as the Product Owner or as a peer member of the Development team. Also, a Product Owner/ Development team member combination will diminish both roles and neither would be performed in a good way.

Development team member and ScrumMaster: The Development team member needs to work with the other Development team members to complete the tasks, realize the Sprint Goal and deliver working software. The ScrumMaster is expected to ensure that there is nothing in the way of this work and to optimize efficiencies and look for ways to increase performance. The ScrumMaster needs to be able to see the forest through the trees; the team member needs to concentrate on the trees themselves. It's nearly impossible to do both things well. Also, a Development team member / Scrum Master combination will diminish both roles and neither would be performed in a good way.

So, in short – do your best to avoid combining any of the three Scrum roles.

8.6 WHAT IS THE ROLE OF MANAGEMENT IN SCRUM?

Scrum does not exist in a vacuum – there is a role for management even in an agile organization using Scrum as a framework. However, how a managerial role is played will change significantly in an Agile organization.

A manager in an Agile organization will focus on building a learning organization and create the best possible environment and conditions for the Scrum team to succeed. This manager will have a coaching style and align the members of the organization by clearly communicating the common purpose of the organization. An Agile manager will also be expected to work with the Scrum Master or Scrum Masters for continuous improvement and maximize efficiency by removing organizational impediments.

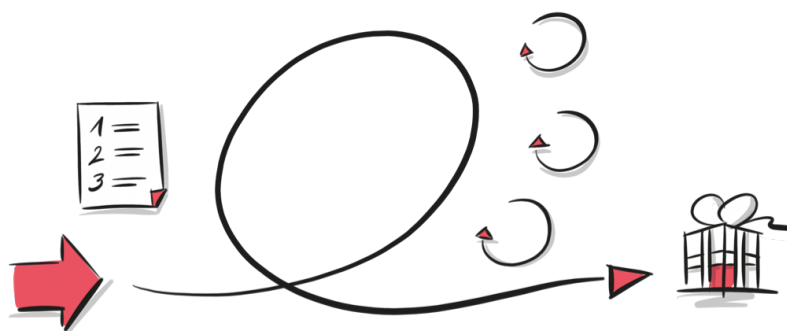
9 SCRUM EVENTS

Scrum prescribes five events. These events are used to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events with a maximum duration, ending either when the maximum time-box is met or sooner if the purpose of the event has been fulfilled.

Except for the Sprint, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include or perform any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

9.1 THE SPRINT

The heart of Scrum is a Sprint, a time-box of one month or less during which a “Done”, useable, and potentially releasable product Increment is created. Sprints have consistent durations throughout a development effort (usually one, two, three or four weeks). A new Sprint starts immediately after the conclusion of the previous Sprint. Sprints contain and consist of the Sprint Planning, Daily Scrums, the Sprint Review and the Sprint Retrospective.



Each Sprint has a goal of what is to be built, a design and flexible plan that will guide building it, the work, and the resulting product Increment. During the Sprint no changes that would endanger the Sprint Goal can be introduced. This does not prevent the scope from being clarified during the Sprint. The Development team may add or drop smaller tasks as they see fit.

Quality is a constant in Scrum so our quality goals do not decrease during the Sprint, even if it means we won't be able to complete a feature. The quality level the Scrum team delivers is documented in their Definition of Done.

How long should our sprints be?

Sprints have a maximum timebox of 4 weeks. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a Sprint Goal at least every calendar month. Sprints also limit risk to one calendar month of cost.

So how long should your Sprints be? There is no right answer, but there are a few things to consider.

Shorter sprints mean faster feedback and more opportunities for learning, both on a product and process level. Longer Sprints might be tempting but they can also conceal underlying problems that should be addressed.

For products like software, marketing campaigns etc, shorter Sprints (1-2 weeks) will work well. If you are working on infrastructure, embedded software including development or modification of hardware the trend is to have longer Sprints of typically 3-4 weeks.

Do not exceed the maximum time-box of one month for the Sprint, it increases risk and reduces feedback and learning. If there is a strong push to have Sprints longer than 4 weeks it's a strong indication that something else need to be changed in the way the organization is working.

The length of the Sprint can be seen as a compromise between how long time the Development team needs to complete something of value to the end customers and how often the Product Owners need to change priorities and goals in order to maximize value. There are a lot of good practices that can be employed by the Development team to shorten the time needed to complete something of value. The Product Owner is in some sense dictate by the market the product operates in. If competition is fierce and the market changes on a weekly basis the Product Owner will need to have the flexibility to change priorities and goals every week.

9.2 SPRINT PLANNING

Each Sprint begins with a Sprint Planning. In this event the Scrum Team collaborates to select and understand the work to be done in the current Sprint.

The entire team attends Sprint Planning. Working from the ordered Product Backlog, the Product Owner and the Development Team Members discuss each item and come to a shared understanding of that item and what is required to complete it consistent with the current Definition of Done. The recommended time for Sprint Planning is two hours or less per week of Sprint duration. Because the event is time-boxed, the success of the Sprint Planning is highly dependent upon the quality of the Product Backlog going in. This is why Product Backlog Refinement is an important Scrum activity.

The Scrum Master ensures that the event takes place, that attendants understand its purpose and that it's kept within the time-box.



The Sprint Planning event covers two topics; 1) What can be delivered in the Increment resulting from the upcoming Sprint? 2) How will the work needed to deliver the Increment be achieved?

Topic One: What can be done this Sprint?

In the first part of the event, the Product Owner presents the focus for the next Increment of the product. This could be a new feature, addressing risks or testing a hypothesis. Based

Cynthia | AXA Graduate

AXA Global Graduate Program

Find out more and apply

redefining / standards



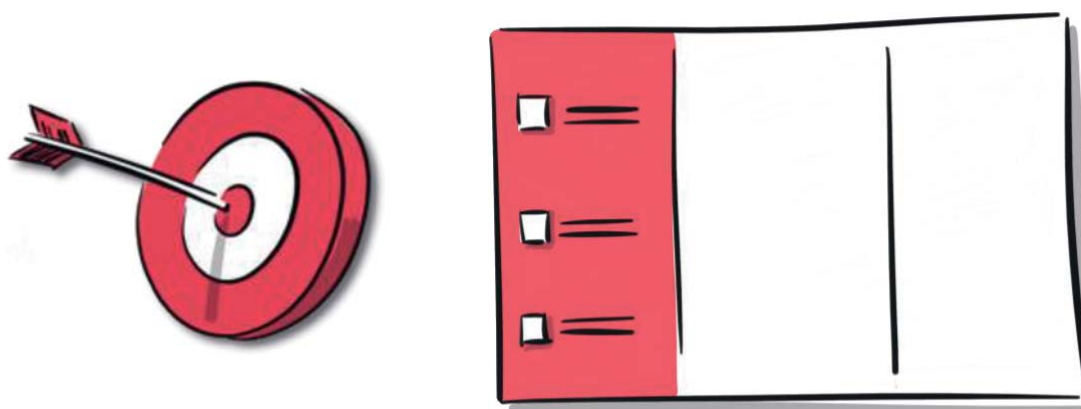
on the goal of the sprint the Scrum team collaboratively crafts a Sprint Goal. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint.

The Sprint Goal is an objective that should be met within the Sprint through the implementation of the selected Product Backlog items. A clear Sprint Goal will also provide guidance to the Development Team on why it is building the Increment.

The number of Product Backlog items to undertake in the Sprint is solely up to the Development Team. To decide how many items to undertake, the Development Team considers the current state of the product Increment, the past performance of the team, the team's current capacity, and the ordered Product Backlog. The Development Team alone decides how much work to take on. Neither the Product Owner, nor any other agency, can push more work onto the Development Team.

Topic Two: How will the chosen work get done?

In the second part of the event, the Development Team collaborates to decide how to produce the next Product Increment in accord with the current Definition of Done. The Development Team usually starts by designing the system and the work needed to convert the Product Backlog into a working product Increment. They do sufficient design and planning to be confident of completing the work during the Sprint. Work to be done in the early days is broken down into small units of one day or less. Work to be done later may be left in larger units to be decomposed later.



The Product Owner can help to clarify the selected Product Backlog items and make trade-offs. If the Development Team determines it has too much or too little work, it may renegotiate the selected Product Backlog items with the Product Owner. The Development Team may also invite other people to attend to provide technical or domain advice.

The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog. By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

Sprint Planning concludes with the Scrum Team coming to a common understanding of the quantity and complexity of what is to be accomplished during the Sprint, and within a rational range of circumstances, expect to complete it. The Development Team forecasts the amount of work they will complete and commit to each other to accomplish it.

Note

When the Development Team forecasts this does not mean that they promise they can get everything done. Software development is complex and unforeseen obstacles may occur.

9.3 SPRINT BACKLOG AND SPRINT GOAL

The Sprint Backlog is a list of all the work that the Development Team identifies as necessary to deliver the Sprint Goal into a “Done” Increment. The Sprint Backlog should have enough detail so that changes in progress can be understood in the Daily Scrum. To ensure continuous improvement, it includes at least one high priority process improvement identified in the previous Retrospective meeting.

The work in the Sprint Backlog should be guided by the Sprint Goal. As the Development team works through their plan of delivering the Sprint Goal they will learn more about the work. The Development Team will modify the Sprint Backlog throughout the Sprint as they learn more. The Development team may add, drop or change work items (often referred to as tasks) as needed to reach the Sprint Goal. This flexibility allows the Sprint Backlog to emerge during the Sprint and continuously incorporates the learning of the Development team into the current plan.

To ensure transparency the Sprint Backlog should be highly visible and represent a close to real-time picture of the work that the Development Team plans to accomplish during the Sprint. The Development Team, being a self-organizing unit, owns the Sprint Backlog and are themselves responsible for monitoring and updating their progress.

If the work turns out to be significantly different from what the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

SPRINT PLANNING QUICK REFERENCE

PURPOSE

Define Sprint goal, choose work from the Product Backlog and to create a realistic plan for the upcoming Sprint.

PARTICIPANTS: The entire Scrum Team

DURATION: Maximum time-box: 2 hours per week Sprint

PREPARATION

- The Product Owner prepares a Sprint Goal and selects the desired scope from the Product Backlog
- This Sprint Goal and scope should consist of a realistically estimate of Product Backlog Items
- The team knows what capacity they have available for the Sprint and what they have been able to deliver in previous Sprints.

IMPLEMENTATION

- The meeting is done in two stages;
 - Topic 1: the what - focuses on Sprint goals and scope
 - Topic 2: the how - focuses on detailed planning

SPRINT PLANNING PART 1

- The Product Owner presents the suggested Sprint Goal and the whole Scrum team collaborates to crafts a realistic Spring Goal securing a sense of whole team ownership to the goal.
- The Product Owner presents the Product Backlog items from the Product Backlog that would achieve the Sprint goal
- The Product Owner and the Development Team agrees on a realistic scope, and the Sprint goal is established.

SPRINT PLANNING PART 2

- The Development Team discuss how it will build this functionality into a “Done” product Increment during the Sprint
- The Development team creates a Sprint Backlog consisting of all the activities necessary to build all the selected Product Backlog items to the current Definition of Done and achieve the Sprint Goal.
- Once this is done the team should forecast if this scope is realistic and something they can commit to. Many teams estimate the Sprint Backlog in hours and compare this to their capacity.
- If the Development Team discovers that the scope is too much, the Development team and Product Owner should collaborate about which items to remove.

RESULT

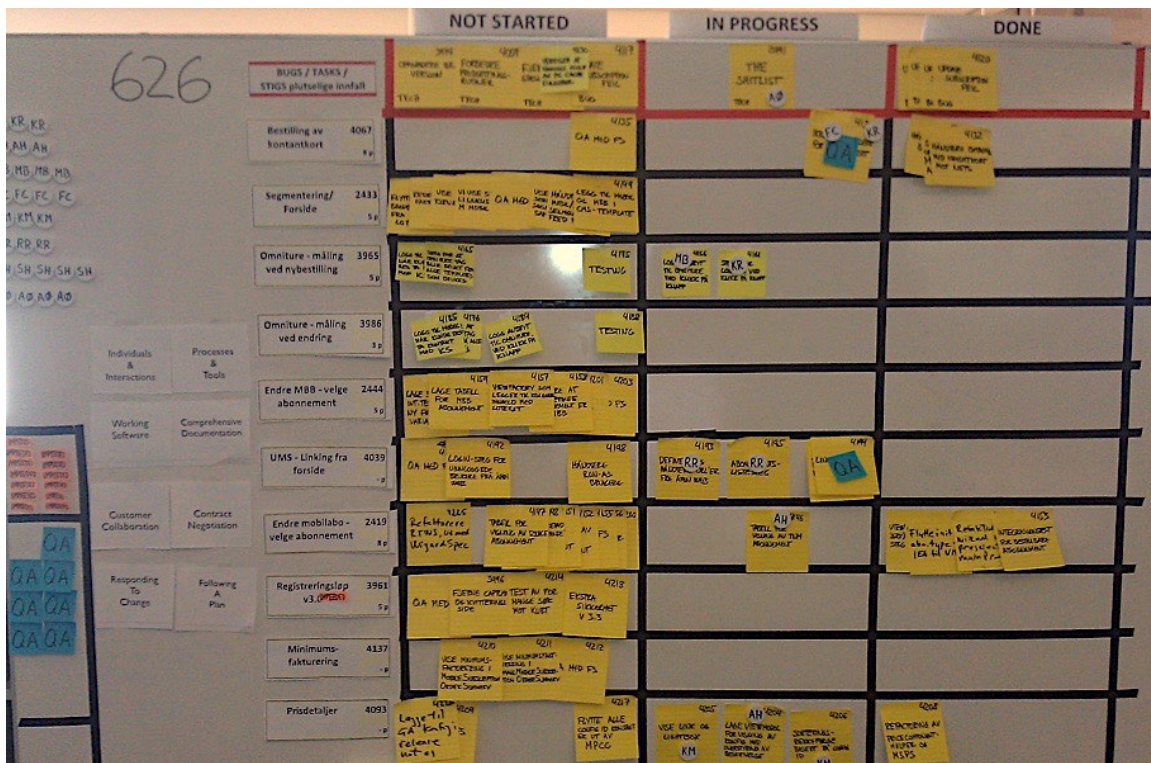
- Sprint Backlog with Sprint Goal

Visual management: The Scrum Board

The Sprint Backlog should be highly visible. This can be achieved using visual management to create an information radiator. This information radiator is referred to as the Task board or Scrum board. The Scrum board is a visualization of the Sprint Backlog.

The Scrum board should be a physical board. It is a "living" artifact that has to be manually maintained by the Development team. An incomplete or stale Scrum board is worthless. A Scrum board should be updated several times a day to be kept healthy. A good Scrum board should be carefully designed with readability and usability in mind. You have a great Scrum board if;

- Team members never complain about having to use it.
- The Daily Scrum happens *against* it.
- Random people that pass by stop to look at it, expressing interest and curiosity.
- Your boss has proudly shown it to his boss.
- You see team members updating it regularly during the day.
- It passes the hallway usability test: a person who has never seen it before can understand it quickly and without explanations



9.4 DAILY SCRUM

The Development Team in Scrum is self-organizing. The Development Team uses the Daily Scrum to inspect that they are on track for attaining the Sprint Goal. They also look at the trend towards completing all the work in the Sprint. The Daily Scrum is held at the same time and place every day of the Sprint. The timebox for the Daily Scrum is 15-minutes.

During the Daily Scrum updates their progress, coordinates and plans work for the next 24 hours (until the next Daily Scrum). This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work.

The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal through the continuous inspection and adaption. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint.



The structure of the meeting is set by the Development Team and can be conducted in different ways. However, the Development team must make sure they focus on progress toward the Sprint Goal. Some Development Teams will use questions, some will be more discussion based. Here is an example of what might be used:

- What did I do yesterday that helped the Development Team meet the Sprint Goal?
- What will I do today to help the Development Team meet the Sprint Goal?
- Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

Note that the guiding questions are focused on the team level and towards achieving the Sprint goal.

The Daily Scrum is for the Development team by the Development Team. It is a communication meeting within the Development Team, to ensure that they are all on the same page. If others are present, the Scrum Master ensures that they do not disrupt the meeting. The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Product Owner should be present and ready to collaborate with the Development team and understand how the Sprint is progressing.

There may be brief clarifying questions and answers, but there is no discussion of any of these topics during the Daily Scrum, there simply is no time. However, many teams or team members meet right after the Daily Scrum to work on any issues that have come up, detailed discussions, or to reorganizes the work as needed to accomplish the Sprint Goal.

The Daily Scrum leads to transparency, trust, and better performance. It provides rapid recognition of problems (impediments), and promotes the Development team's self-organization and self-reliance. It also improves communication, eliminate other meetings, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.



- The number 1 MOOC for Primary Education
- Free Digital Learning for Children 5-12
- 15 Million Children Reached

About e-Learning for Kids Established in 2004, e-Learning for Kids is a global nonprofit foundation dedicated to fun and free learning on the Internet for children ages 5 - 12 with courses in math, science, language arts, computers, health and environmental skills. Since 2005, more than 15 million children in over 190 countries have benefitted from eLessons provided by EFKI. An all-volunteer staff consists of education and e-learning experts and business professionals from around the world committed to making difference. eLearning for Kids is actively seeking funding, volunteers, sponsors and courseware developers; get involved! For more information, please visit www.e-learningforkids.org.

Try this for Daily Scrum

Many the Development teams find it hard to conduct great Daily Scrums. Here are some tips that might help. Remember to try one thing at the time! If you would like to experiment with several of these tips introduce them over time and use the Sprint Retrospective to find out if they are useful for the team.

Use an object (a pen, ball or as one Development team I worked with a stuffed toy salmon in full size) to manage who is speaking. Only the person with the object can talk and when the person is finished, he throws the object to someone else in the team.

Have the Daily Scrum just before lunch. The Development can decide for themselves when they would like to have their Daily Scrum. However, it means stopping what you are doing and attending the Daily Scrum. Since people tend to get to work at different time and also leave at different times, many teams find the morning or afternoon less than ideal, and it interrupts with your work. One way to deal with this is to have the Daily Scrum just before lunch. It definitely helps keeping it within the 15 minutes (don't mess with lunch...), team member will be interrupted by lunch anyway and it also have the positive side-effect that the team have lunch together.

Be future oriented. People have a tendency to focus on what is closest in time and I have observed many, many Daily Scrums where the team members spend almost all the time talking about what they have done since the last Daily Scrum and what they are doing now. However, team members and Development teams collectively often fail to talk about what they are planning to do next, until the next Daily Scrum. We can't change what has already happened but we can influence and plan for the future. A Development team should use the majority of the time during the Daily Scrum to collectively create a plan for the next 24 hours maximizing the probability for achieving the Sprint goal. If you find you Development team is not focused on the future try to have each team member start their contribution in the Daily Scrum with: *"Until the next Daily Scrum I plan to....."*

One-minute retro on the Daily Scrum. Just after they Daily Scrum the Development team collectively should answer a few questions:

1. Did we inspect our progress towards the Sprint goal?
2. Did we inspect the trend towards completing all the work in the Sprint?
3. Do we need to replan?
4. Are all team members working on something that brings us closer to the Sprint goal?

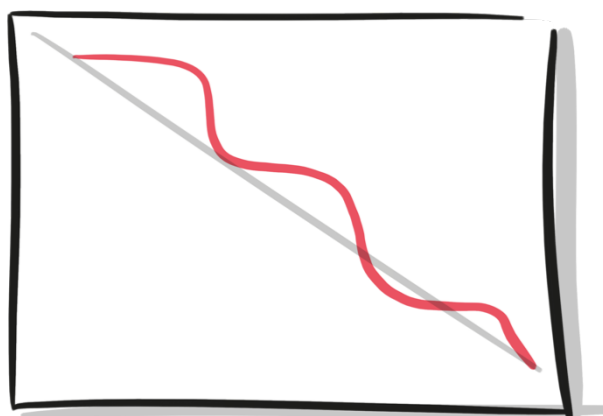
Dependent on the answers to these questions the Development team might meet just after the Daily Scrum to make necessary adjustments.

9.5 MONITORING SPRINT PROGRESS

The Development Team is responsible for monitoring their progress towards the Sprint Goal and trend towards completing all the work in the Sprint. At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be summed. The Development Team tracks this total work remaining at least for every Daily Scrum to forecast the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Development Team can self-manage its progress.

Scrum does not dictate how the Development team should go about monitoring and visualizing their progress. However, many Development Teams do this by using a Sprint Burndown Chart.

In a Burndown Chart the Development Team updates the estimated remaining work in relation to remaining capacity every day. This way everybody can easily inspect the realism of the current Sprint. Below is an example of a Sprint Burndown chart.



The Y-axis represent the scope of the work, usually measured in hours or number of tasks. The X-axis is time, from the start of the sprint until its conclusion. The straight gray line in the figure above shows the Development Team's total remaining capacity from the first day of the sprint, down to zero remaining capacity on the last day of the sprint. The red wavy line shows estimated remaining work during each day of the Sprint. We are only concerned about estimated remaining work because this is the information the Development teams needs to be able to inspect their own progress and evaluate if they can achieve the Sprint goal.

The Development team will update the Sprint Burndown in connection with Daily Scrum.

DAILY SCRUM QUICK REFERENCE

PURPOSE

Coordinating the team and create a plan for the next 24 hours (until the next Daily Scrum). Inspect progress towards the Sprint Goal and the trend towards completing all the work in the Sprint.

PARTICIPANTS: Development Team. It is recommended that the Scrum Master and Product Owner observes the Daily Scrum.

PREPARATIONS

- All Development Team Members have thought through what they did since the last meeting, what they plan to do before the next meeting, and what they might see as problems or obstacles.
- Daily Scrum is held at the same time and same place every day

DURATION: Maximum of 15 minutes

IMPLEMENTATION

- All members of the Development Team take turns speaking and explaining what they have done, what they plan to do and if they see anything threatening the Sprint goal;
- Problems and obstacles are noted
Alternatively: «walk-the-wall» where the team reviews the active tasks instead of basing it on individuals.
- Realism of the Sprint is evaluated

Tip

There is no time to solve problems in this meeting. If a discussion arises it is important to maintain discipline and stop it, and instead follow up on it after the Daily Scrum.

RESULT

- The Development Team is coordinated
- The Sprint Backlog is updated
- Obstacles have been identified
- Realism of the Sprint has been evaluated; If the remaining work is too little or too much the scope is re-evaluated. The Product Owner has the final say when it comes to adding or removing.

9.6 SPRINT REVIEW

At the end of the Sprint, the Scrum Team and stakeholders review the output from the Sprint. They inspect the Increment and collaborate about what was done in the Sprint. This is an informal meeting to take a look at where we are. The presentation of the Increment is intended to elicit feedback and foster collaboration. The presentation of the Increment should be done as a demonstration of the working product – there are no slides in this meeting. It's the responsibility of the Product Owner to make sure that key stakeholders attend the Sprint review. The recommended time box for the Sprint Review is one hour per week of Sprint duration.

A well facilitated Sprint review starts with the Product Owner explaining what Product Backlog items have been “Done” and what has not been “Done” during the Sprint. The Product Owner is expected to have daily interaction with the Development team so there are no surprises for the Product Owner in what Product Backlog items have been completed and if there were any items the Development team wasn't able to complete according to the Definition of Done. The Increment is kept transparent when the Development team accomplish all their work according to the Definition of Done. Then the Development team will demonstrate the Increment and show the work that met the Definition of Done. During the demonstration the Development team will answer questions. After this demonstration (which is a collaborative inspection) all attendees (the Scrum team and stakeholders) should have a shared and clear understanding of the state of the Increment.



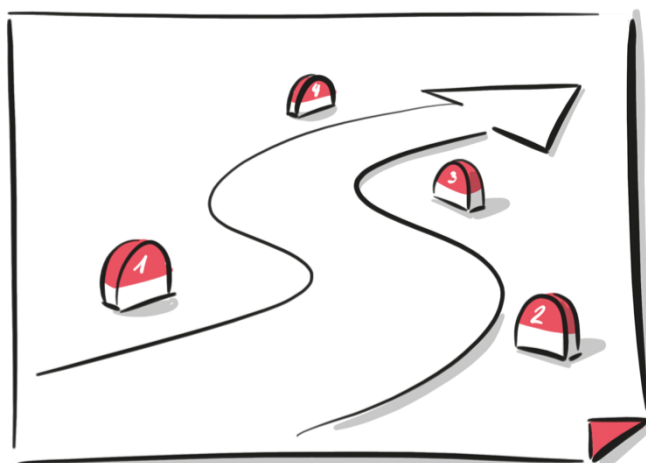
The Product Owner will then briefly discuss the Product Backlog as it stands. The purpose of this is to make sure there is a shared understanding of the planned future development of the product. All participants then collaborate on how the marketplace or potential use of the product might have changed and suggest changes or additions to the Product Backlog. The Product Owner, stakeholders and Development team are collaboratively adapting the planned future development of the product based on our new insights about the Increment, the market and feedback from the demonstration. Existing Product Backlog Items might also be dropped if objectives have already been met.

Everyone should have input at the Sprint Review, but the Product Owner makes the final decisions on what suggestions will be added to the Product Backlog. The Product Owner must keep in mind that there is almost always more request, wishes and demand than there is available development capacity for the product.

The Product Owner is also responsible for keeping track of progress towards a release, a project or a budget constraint. After all attendees have collaborated adapting the Product Backlog the Product Owner will review the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product. In empirical product development the Product Owner will use progress to date to project likely target and delivery dates. This information should be updated at least every Sprint Review and is made transparent to all stakeholders.

Based on projected progress and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. In this way the Sprint Review also provide valuable input to the subsequent Sprint Planning.

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.



Use a burn-up chart to visualize progress

The Release Burn-up chart is a visual management tool to make the progress towards a release transparent. It is a graphic representation that shows how much work is completed each sprint, and can also be used to develop both time-based or scope-based forecasts. Release burn-up charts have proven useful. However, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making. Hence the projection done based on burn-up chart is a prognosis and only that.

To construct a burn-up chart we need to have a measure of the work completed each Sprint. The amount of work completed each Sprint is often referred to as velocity. Velocity is a useful long-term measure of the amount of work completed per sprint, but not an exact prediction of how much work will be completed in each sprint. Velocity is measured in the same units the Development Team use to estimate Product Backlog items. Notice that only work that is completed in accordance with the Definition of Done at Sprint Review will count against the velocity. I.e. if a Product Backlog Item was estimated to 8 story points and the Development is 90% complete with this PBI in time for the Sprint Review it will count as 0 - zero – in velocity.

This strict definition helps us focus on quality and make sure the work flows into the hands of customers or users that will provide real and valuable feedback. Code or pieces of a product not yet released is a form of inventory and should be avoided. Hence it is a good practice to not give credit for partially done work.

For a full walkthrough of the burn-up chart please refer to: <http://scrummaster.no/2019/burn-up-chart/>

SPRINT REVIEW QUICK REFERENCE

PURPOSE

Inspect the new Increment and demonstrate Product Backlog items that meet the Definition of Done. Getting feedback from stakeholders and adapt the Product Backlog. Update progress release plan and budget.

PREPARATIONS

- Product Owner invites the most relevant stakeholders
- Product Owner has a clear understanding what was actually completed in the Sprint in accordance with the definition of Done
- Development Team is ready to demonstrate the Done functionality

DURATION: Maximum time-box of 1 hour per week Sprint

IMPLEMENTATION

- Product Owner reviews what was Done and optionally what was not Done during the Sprint.
- Development Team demonstrate the functionality that is “Done”. Stakeholders or users may try the product themselves. Everyone in the Scrum team answers questions.
- Product Owner notes comments, suggestions for improvements, ideas and feedback from stakeholders and participants in the event
- After the demonstration the Product Owner reviews the Product Backlog as it stands. Everyone in the meeting provides viewpoints on what they want to do next.
- Product Owner reviews plan, budget and progress towards the next release date

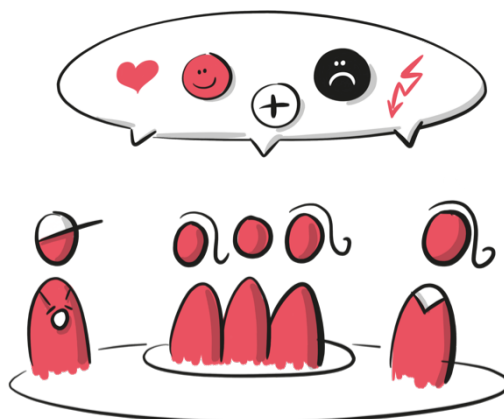
RESULT

- The entire Scrum Team and stakeholders have a common understanding of the results of the Sprint and status of the product
- The Scrum Team have learned something about the product, market and/or customers/users together with the stakeholders
- Product Backlog has been adapted to maximize the competitiveness of the Increment given future development
- Probable Product Backlog items for the next Sprint have been identified

9.7 SPRINT RETROSPECTIVE

At the end of each Sprint, the Scrum Team meets for the Sprint Retrospective. The purpose is to review how things went with respect to the process, the relationships among people, the tools, practices and whatever else the Scrum team needs to discuss. The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself, an area for team self-improvement and continuous learning. The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. The recommended time box for the Sprint Retrospective is 45 minutes per week of Sprint duration.

The Sprint retrospective is a closed event for the Scrum team. The Scrum Master ensures that the meeting is positive and productive, keeping the focus on how we can improve the system we're a part of. The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of "Done", if appropriate and not in conflict with product or organizational standards



By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself.

Improvements within the control of the Development Team should be implemented by the Development themselves. The Scrum team will also identify improvements related to the organizational context, policies and practices. It's the responsibility of the Scrum Master to address these impediments, by either removing or eliminating them all together. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.



Rinse, repeat.

The Scrum cycle repeats from here, for every Sprint.

RETROSPECTIVE TECHNIQUES

The Scrum Master is responsible for keeping the Sprint Retrospective positive and productive. It is a good idea to vary how these meetings are conducted to ensure active participation from the entire team. Below you can find some links to pages with several hundred techniques you can use as a part of the Sprint Retrospective.

[Retrospectivewiki.org](https://retrospectivewiki.org) - wiki with ready-made plans for retrospectives, tools, tips & tricks and more.

<https://retromat.org/en/> - online generator that creates suggestions for retrospective plans. You can also swap elements and create your own

Tip

Use this governing principle for Sprint Retrospectives;

“Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand.”

– Norm Kerth, Project Retrospectives

Tip

To reduce many improvement ideas (e.g. in the form of post-its) in a brainstorming session down to a few actionable items, try this:

- Everyone helps grouping notes that are related. Name the groups.
- Use “dot-voting”: each participant gets a small number of dots (use a marker) that they allocate to the items they find most important.

Use techniques like 4 L’s to categorize the experiences from the last Sprint. Feel free to use a flip-over sheet for each category, and let the participants attach post-its to the four sheets:

- **Liked:** This worked great and I would like to continue with it
- **Learned:** Moments of realization and other newly gained knowledge from the Sprint
- **Lacked:** Wish we had this! Specific examples.
- **Longed for:** Long-term wishes, maybe not entirely realistic.

SPRINT RETROSPECTIVE QUICK REFERENCE

PURPOSE

For the Scrum team to inspect itself and create a plan for improvements. To learn from experience to improve the efficiency of both the team and the organization.

PREPARATION

- Scrum Master is responsible for the event
- The whole Scrum Team should participate

DURATION

- Maximum timebox of 45 minutes per week Sprint

FACILITATION

- Scrum Master is responsible for facilitating the event. Facts from the Sprint is used as input.
 - Example: Feedback from the Sprint review, number of support cases, blocked cases, events, instances where workload was underestimated, etc.
 - Everyone contributes
- Everyone then contributes with their experience of the Sprint; What worked, what would they like to improve upon. Use different techniques to engage everyone in the team.
- Reduce the improvement suggestions down to a few prioritized improvement (minimum 1, recommended max 4)
- Agree on who does what, Scrum Master follows up

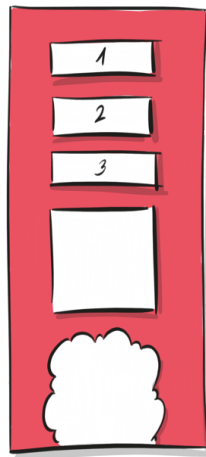
RESULT

- Minimum one improvement the Development team will take on in the next sprint
- A small number of actions with a clearly identifies responsible person visible
- A list of identified impediments
- Shared knowledge - every team member hopefully leaves the retro-meeting a little bit wiser

10 SCRUM ARTIFACTS – QUICK REFERENCE

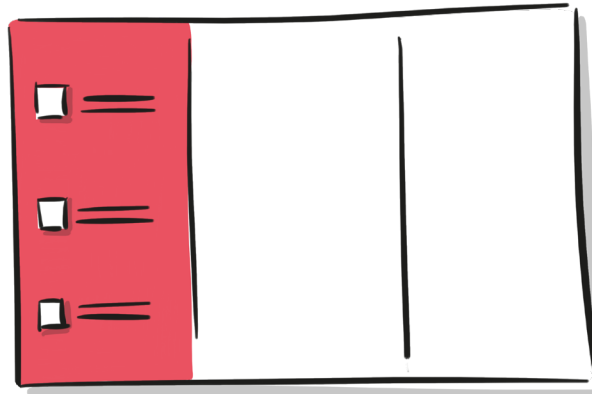
Scrum's artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

10.1 PRODUCT BACKLOG



The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

10.2 SPRINT BACKLOG



The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a “Done” Increment.

10.3 INCREMENT

The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. At the end of a Sprint, the new Increment must be “Done,” which means it must be in useable condition and meet the Scrum Team’s definition of “Done”. An increment is a body of inspectable, done work that supports empiricism at the end of the Sprint.



11 REFERENCES AND FURTHER READING

This book is a representation of my knowledge and experience gathered over the last 15 years. I have read a lot of books, articles, blogs through those years so I may not have been able to recall where I read something first. Notwithstanding, I've made an honest attempt to reference the sources I know have inspired me and that I have used when writing this book.

Thanks to Holger Nils Pohl and the WorkVisual Institute for the amazing illustrations.

<https://www.agilealliance.org/agile101/>

Adkins, Lyssa. (2010). Coaching Agile Teams. Addison-Wesley Signature Series.

Alistair, Cockburn. <https://alistair.cockburn.us/information-radiator/>

Coplien, James. (1994). Borland Software Craftsmanship: A New Look at Process, Quality and Productivity.

Cohn, Mike (2010). Succeeding with agile. Software development using Scrum. Addison-Wesley Signature Series.

Cohn, Mike (2006). Agile estimating and planning. Prentice Hall Professional Technical Reference.

Derby, Esther and Larsen Diana (2006). Agile Retrospectives. Making good teams great. The Pragmatic Bookshelf.

Galsworth, Gwendolyn. (2005). Visual Workplace/Visual Thinking. Visual-Lean enterprise Press.

Kim, Gene. Debois, Patrick, Willis, John. Humble, Jez. Allspaw, John. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press, Portland, Oregon.

Kim, Gene. Behr, Kevin. Spafford, George (2018). The Phoenix Project (5th anniversary edition). IT Revolution Press, Portland, Oregon.

Fowler, Martin (2006). Writing The Agile Manifesto. <https://martinfowler.com/articles/agileStory.html>.

Larman, Craig & R. Basili, Victor. (2003). Iterative and Incremental Development: A Brief History. Computer. 36. 47 - 56. 10.1109/MC.2003.1204375.

Larman, Craig & Vodde, Bas. Large Scale Scrum (LeSS). www.less.works

Liker, Jeffrey K (2004). The Toyota Way. 14 Management Principles. McGraw-Hill.

Lacey, Mitch. (2012). The Scrum Field Guide: Practical Advice for Your First Year. Addison-Wesley Professional.

Poppendieck, Mary and Poppendieck, Tom (2003). Lean Software Development. An Agile Toolkit. Addison-Wesley.

Poppendieck, Mary (2001). Lean Programming. <http://www.drdoobbs.com/lean-programming/184414734>

Schwaber, Ken. (2004) Agile Project Management with Scrum. Microsoft Press.

Schwaber, Ken and Beedle, Mike. (2001). Agile Software Development with Scrum (Series in Agile Software Development). Pearson.

Scrum Alliance (2012). Agile Atlas Core Scrum.

The New New Product Development Game, Hirotaka Takeuchi & Ikujiro Nonaka. January 1986 issue of Harvard Business Review.

The Scrum Guide. www.scrumguides.org – Ken Schwaber and Jeff Sutherland

Quesada Allue, Xavier. The Visual Management Blog. <http://www.xqa.com.ar/visualmanagement/2009/02/visual-management-for-agile-teams/>