

Scrum

Certified Scrum Product Owner course handouts



Our high quality Agile coaching and training services support our customers in delivering better products faster!

www.iLean.be | info@ilean.be | +32 (0) 3 443 12 03

Table of contents

Scrum framework	3
What is Scrum?.....	4
Scrum framework overview	5
Scrum roles	6
Scrum rituals	7
Scrum artifacts.....	8
Scrum and Agile principles	9
Scrum principles	10
Scrum values.....	13
History of Scrum	14
Agile	15
Product Backlog and user stories	16
A 'ready' Product Backlog	17
User stories	18
INVEST in your user stories	19
User story pitfalls	20
Agile estimation and planning	21
Relative estimations	22
Velocity	24
The release burndown graph.....	25
The Scrum Rituals	26
Product Backlog Refinement	27
Sprint Planning	28
Daily Scrum	30
Sprint Review	32
Sprint Retrospective	34

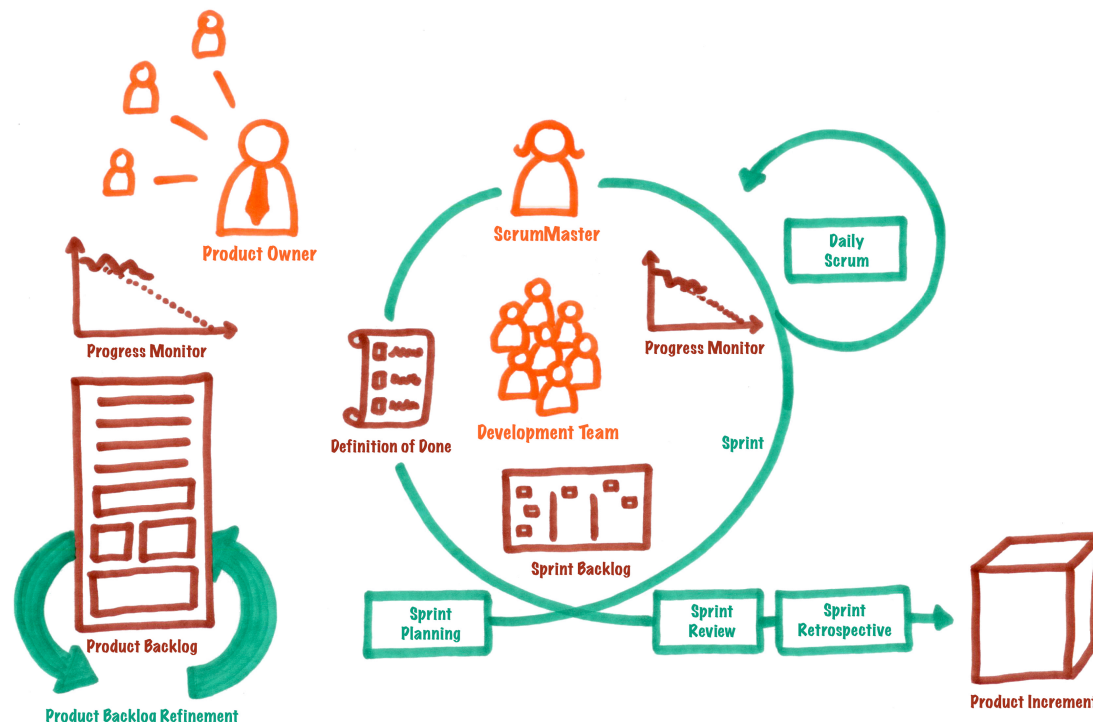
Scrum framework

What is Scrum?

Scrum is a **framework** for building products. The goal of Scrum is to do this with a self-organizing team using short, time-boxed iterations that each deliver a potentially shippable increment of the product. This way, Scrum maximizes control, flexibility, quality and return on investment.

The Scrum framework defines a limited, well-balanced set of roles, rituals and artifacts that work together according to certain rules. Next to these, Scrum requires, and at the same time enables a specific set of values and principles.

Scrum framework overview



Scrum starts with the need from stakeholders for a certain product. The **Product Owner** crafts a vision for the product and collects the product goals and requirements in the **Product Backlog**. He is responsible for maximizing the delivered value of the product, taking into account time, budget and other constraints. He does so by continuously **refining the Product Backlog** together with the Development Team.

The **Development Team** is a cross-functional and self-organizing team that delivers releasable, valuable increments of the product in short, time-boxed iterations called **Sprints**. The **ScrumMaster** acts as a guide and coach for the team and organization, supporting them in implementing Scrum well and removing all impediments. The Product Owner, ScrumMaster and Development team together are called the **Scrum Team**.

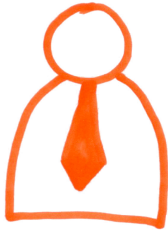
A Sprint starts with **Sprint Planning**. This is a time-boxed ritual in which the Development Team pulls work from the Product Backlog into the Sprint, taking into account the Product Owner's priorities. The outcome of Sprint Planning is the **Sprint Backlog**, which is a plan for the Sprint that contains the team's forecast about the **Product Increment** that will be delivered in this Sprint. The Scrum Team uses the Definition of Done to plan the work and to assess whether a Product Backlog item is truly done and provides business value.

Every day, the Scrum Team gathers for a short, time-boxed meeting called **Daily Scrum** to check the progress, get problems out and assure reaching the Sprint goal. The Scrum team uses **progress monitors** to track and visualize the progress of their work.

At the end of the Sprint the Scrum Team organizes the **Sprint Review** meeting to presents the Product Increment the Product Owner and the stakeholders. And to generate feedback about the product and discuss the progress on the product. After this, the Scrum team will look back on the Sprint and evaluate their process in the **Sprint Retrospective**, resulting in a actionable improvement plan for the next Sprint.

Scrum roles

Scrum defines three different roles: the Product Owner, the ScrumMaster and the Development Team. Together they are called the Scrum Team.



Product Owner

- ✓ Responsible for maximizing product value within time and budget constraints.
- ✓ Represents customers, users and stakeholders.
- ✓ Manages and orders the Product Backlog.



Development Team

- ✓ A cross-functional team including all necessary people (skills and knowledge) to deliver 'done' Product Increments.
- ✓ A self-organizing team
- ✓ Responsible for accomplishing the Sprint goal and delivering a 'done' Product Increment.



ScrumMaster

- ✓ A servant leader supporting the Scrum Team and the organization in implementing Scrum well.
- ✓ Responsible for assuring that impediments are brought up and resolved as soon as possible.

Scrum rituals

Scrum defines four timeboxed meetings:

Sprint Planning

At the start of a Sprint, the Scrum Team meets to forecast and plan the Sprint. The Scrum Team members decide what and how much work is pulled from the Product Backlog into the Sprint Backlog and agree on how to get this work done.

Daily Scrum

Every day, the Development Team members gather for a short (max. 15 minutes) meeting. They discuss the progress of the Sprint, possible impediments and what is needed to reach the Sprint goal.

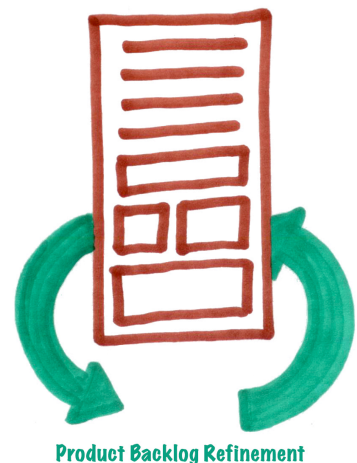
Sprint Review

At the end of the Sprint, the Scrum Team and interested stakeholders meet to review the delivered Product Increment. The Development Team demonstrates the Product Backlog items and interacts with the stakeholders to get feedback. The Product Owner accepts the 'done' work.

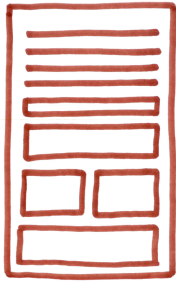
Sprint Retrospective

Right after the Sprint Review, the Scrum Team meets to discuss what went well and wrong during the Sprint. They agree on the most important improvements and define concrete actions and agreements to execute these in the upcoming Sprint(s).

Product Backlog Refinement is not a timebox but an ongoing activity owned by the Product Owner but executed by the entire Scrum Team. It serves to prepare the Product Backlog for the upcoming Sprint(s). This includes adding, splitting, detailing, estimating, ordering etc. of Product Backlog items to make sure the Product Backlog is ready for the next Sprint Planning.



Scrum artifacts



Product Backlog

An ordered list of Product Backlog items that represent all requirements, needs and idea's for the product. Every item must represent a (business) valuable piece of the product. The Product Backlog is the only source of work for the Development Team.



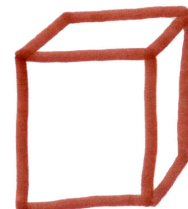
Definition of Done

An agreement between Product Owner, Development Team and stakeholders about what it means for any Product Backlog item to be considered 'done'. Definition of Done does not define the specific (functional/technical) acceptance criteria for individual Product Backlog items, but is a general checklist, applicable for all Product Backlog items.



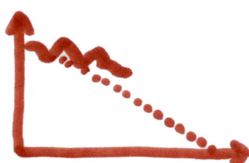
Sprint Backlog

The list of refined Product Backlog items that are selected to be delivered in the current Sprint, together with the team's plan for accomplishing the work. This plan often includes a task breakdown of the Product Backlog items.



Product Increment

A potentially shippable increment of the product. Every Sprint should result in a new Product Increment that meets the Definition of Done.



Progress Monitors

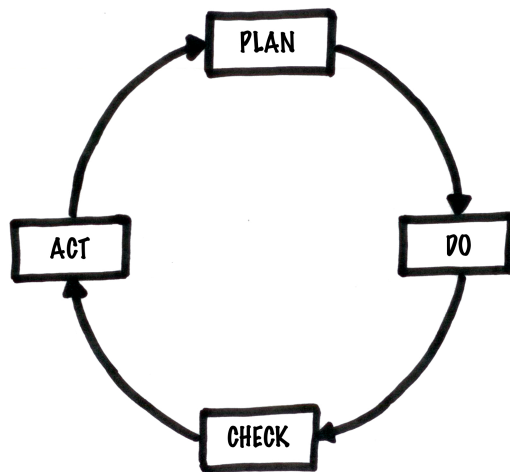
Any artifact created by the Scrum Team to assure transparency about the status and progress of the product. Typical Progress Monitors include task boards and burndown charts on Sprint level and release burndown graphs on product level.

Scrum and Agile principles

Scrum principles

Inspect and adapt (empiricism)

Scrum is an empirical process, meaning it derives knowledge from observation and experience instead of theoretical planning and assumptions. To implement this **inspect and adapt** principle, Scrum is based on the PDCA cycle, also known as the Deming quality cycle. This is an iterative method for assuring control and continuous improvement when managing processes or products.



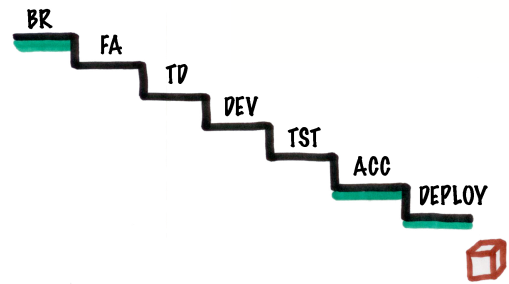
- ✓ **PLAN** Define the goal and expected outcomes and decide how (process) these will be delivered.
- ✓ **DO** Execute the work according to the plan.
- ✓ **CHECK** Measure and analyse the delivered results and compare with the initial plan.
- ✓ **ACT** Investigate the plan, results and process to determine improvements. Implement these improvements in the next cycle.

Every Sprint in Scrum is one run through the PDCA cycle. Sprints are short and timeboxed to assure that assumptions made in planning are checked with reality as early and frequent as possible, leading to improved quality and results.

Iterative and incremental development

Plan-driven

A lot of organizations use more traditional, plan-driven approaches for building products, typically referred to as 'waterfall' approaches. The underlying assumption of plan-driven work is that both requirements as technology are stable and predictable upfront.



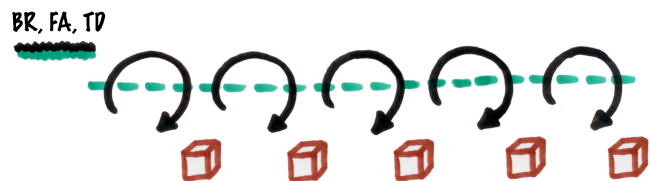
Characteristics:

- ✓ Strictly separated phases, eg. all analysis must be done before development can start. Only 1 run through the PDCA cycle.
- ✓ Customer is only involved early and late in the project (requirements and acceptance testing).
- ✓ Some of the most risky parts, such as integration and testing, are only covered late.
- ✓ Business value (releasable product) is only delivered at the end of the project.

Adaptive

When not all cause-effect relationships can be predicted upfront, it is better to use an empirical and adaptive approach such as Scrum. Every Sprint is a short PDCA cycle that delivers a small but working product increment.

This allows the Scrum team to inspect and adapt very frequently, by evaluating product increments and project progress and by improving the process every single Sprint.



Characteristics:

- ✓ Frequent PDCA cycles including all necessary work (analysis, development, testing) for delivering a working product increment.
- ✓ Continuous (daily) collaboration with the customer is required.
- ✓ Frequent deliveries of working product increments, providing early business value and increasing return on investment.
- ✓ Because of minimal upfront planning, changes in product requirements are easy and cheap.

Remark:

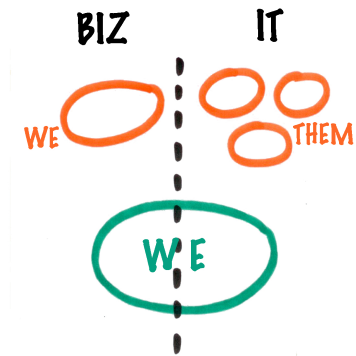


When using an adaptive approach, it is still sensible to build an initial plan for your product. The key to success is to keep the upfront planning and analysis as minimal as possible, and to be prepared for future changes.

Team and Collaboration

Scrum insists that the Development Team contains all the necessary knowledge and skills to deliver 'done' product increments every Sprint. This means team members include people from different departments (typically including business and IT) and with different functions (analysts, engineers, testers, ...).

Although every person might have its own specialized skills and knowledge, the team has one shared goal: collaborate to deliver product increments as effective and efficient as possible.



Self-Organization

The Development Team is a self-organizing team. This means that the team members get to take all the necessary decisions and actions about how to execute the Sprint and how to deliver the next 'done' product increment. No manager, nor the ScrumMaster is micro-managing or command-and-controlling the team.

Self-organization needs clear boundaries and these are provided by the Scrum framework: the Sprint, Definition of Done, Product Backlog, the Product Owner's priorities and acceptance criteria, etc.

Self-organization is not 'do whatever you want' or allowing cowboy-style work. It requires discipline, respect, transparency, communication and collaboration.



Scrum values

The below values are essential in Scrum. An organization or team that wants to implement Scrum needs to understand and embrace these values. Scrum requires these values, and at the same time Scrum facilitates to live these values stronger and more profound.

Focus

Focus is essential for combining quality and speed. By focusing on the most important items only, the Scrum Team collaborates better, produces higher quality products and delivers value sooner.

Openness

A Scrum team provides a transparent and honest view on the status and progress. Concerns and impediments are openly indicated so they can be addressed. This openness pushes teams to better collaboration and increased efficiency.

Commitment

A Scrum team learns how to make trustworthy commitments as a team. This boosts both teamwork and results. By having control over its own destiny, a Scrum team becomes more committed to success.

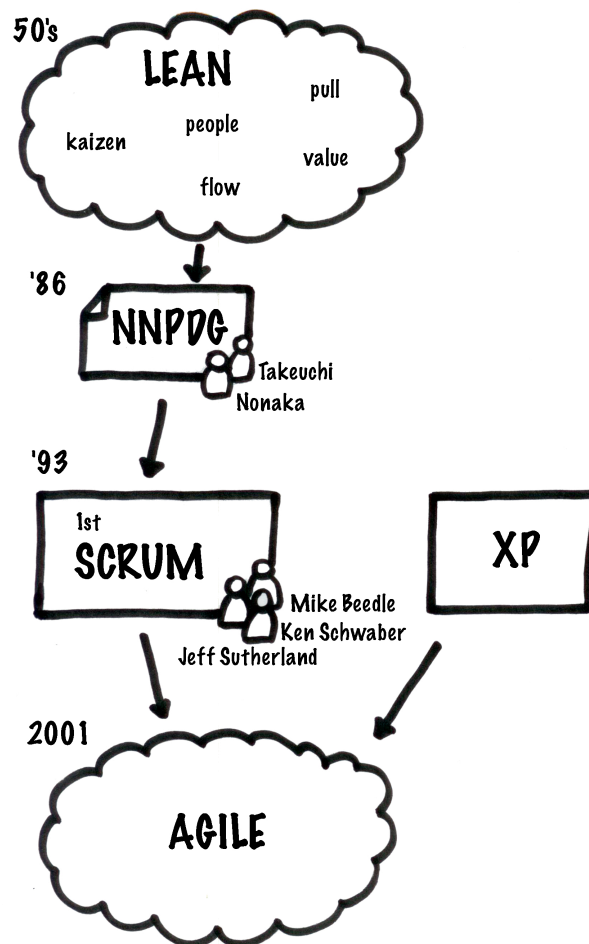
Respect

Excellent teamwork and a transparent environment require respect between members of the Scrum Team. Team members will respect each other and help each other to be worthy of respect.

Courage

Continuously improving, transparency and excellent teamwork requires courage and also provides the courage needed to undertake greater challenges.

History of Scrum



The history of scrum started in 1986 with a Harvard Business Review article “The New New Product Development game” (NNPDG) by Takeuchi and Nonaka. This article was based on the lean principles developed in the car manufacturing world (Toyota). This Lean philosophy was introduced by Toyota with the Toyota Production System as an alternative for the mass productions systems in use at that time.

In the NNPDG article the contrast is made between a ‘relay race’ approach to product development and the holistic or ‘rugby’ approach. The article describes how passing the ball back and forth as a team to go the distance may better serve today’s competitive requirements.

It was only in 1993 that Jeff Sutherland and his team at Easel translated these ideas into the software development practice and implemented the first Scrum. Ken Schwaber and Mike Beedle documented the Scrum process in “Agile Software Development with Scrum”

In 2001 some thought leaders in the software industry met on a conference in Utah to discuss lightweight development methods. They introduced the term Agile and came up with the Manifesto for Agile Software Development that defines their common values and principles.

Agile

The Agile Manifesto defines a set of values and principles. See www.agilemanifesto.org.

Individuals and interactions over processes and tools

- ✓ Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

- ✓ The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

- ✓ The best architectures, requirements, and designs emerge from self-organizing teams.

- ✓ Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

- ✓ Working software is the primary measure of progress.

- ✓ Simplicity - the art of maximizing the amount of work not done - is essential.

- ✓ Business people and developers must work together daily throughout the project.

Working software over comprehensive documentation

Customer collaboration over contract negotiation

- ✓ Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

- ✓ Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

- ✓ Continuous attention to technical excellence and good design enhances agility.

- ✓ At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

- ✓ Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Responding to change over following a plan

Product Backlog and user stories

A 'ready' Product Backlog

The Product Owner must assure that the Product Backlog is healthy and 'ready' at all times. This allows the Development Team to pull the right (most valuable) Product Backlog items into the (next) Sprint and get them done within the Sprint.

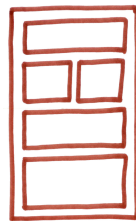
A good Product Backlog is DEEP:

Detailed Appropriately



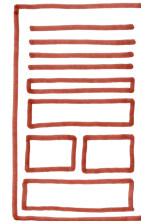
Product Backlog

Too many small and detailed Product Backlog items. No overview and wasteful upfront analysis.



Product Backlog

Good initial product breakdown into high-level items. But no sprintable, small items.



Product Backlog

Correct level of detail: top items are small and sprintable, less important items are larger and need more investigation later.

Emerging

The Product Backlog is a living document, that is continuously updated and improved to define the most valuable product and provide the right information at the right time.

Estimated

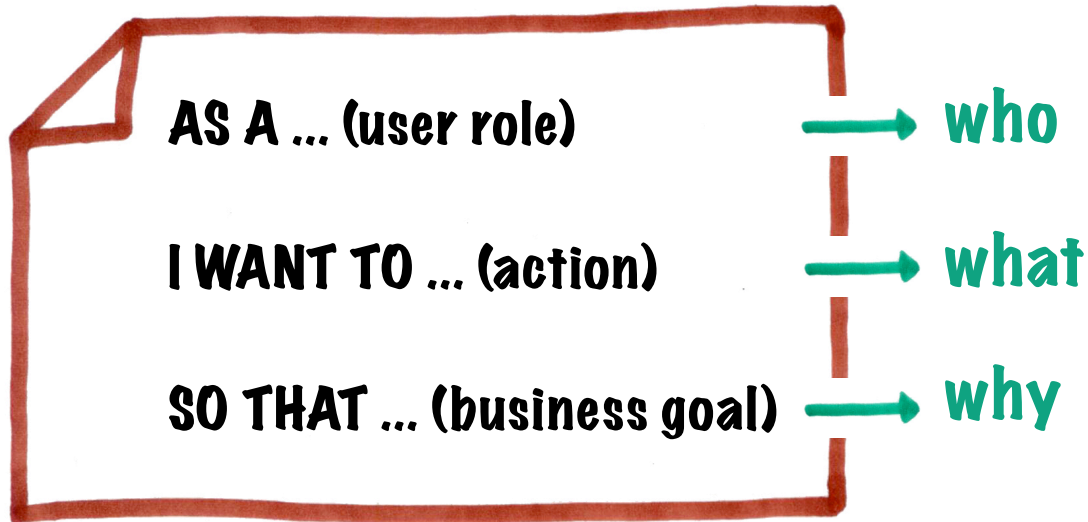
The size or cost of all Product Backlog items should be estimated, to allow the Product Owner to take the right decisions on priorities and project/product planning. And to make sure the top items are small enough to be pulled into a Sprint.

Prioritized

At all times, the Product Owner must assure that the Product Backlog is ordered to build the most valuable product possible.

User stories

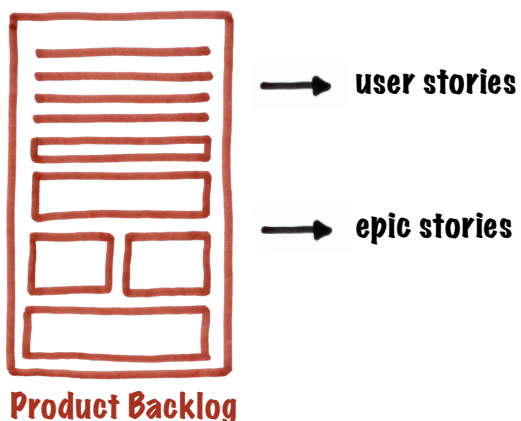
Scrum Teams very often use user stories as Product Backlog items, representing the user requirements or product features. User stories are written in a specific format:



User stories serve different goals:

- ✓ Describe a product feature or user requirement
- ✓ Shared understanding and agreement between business / customers and Development Team
- ✓ A trigger for conversation and just-in-time analysis and design
- ✓ A planning item

Epic versus user stories



The user story format can be applied on any level of detail, describing large and high level requirements but also small and detailed ones. Big, coarse-grained user stories are often called epic stories. They later need to be split into smaller, fine-grained user stories that can be pulled into a Sprint. A good Product Backlog contains both sprintable, small user stories as larger epic stories that are not yet sprintable and need further refinement.

INVEST in your user stories

When writing user stories, it is good to keep the INVEST acronym in mind to come to good, well-formulated user stories.

I**NDEPENDENT** User stories should be as independent of each other as possible, allowing more flexible planning and implementation.

N**EGOTIABLE** A good user story is not a detailed contract, but allows co-creation and negotiation of the details. It should capture the essence needed for estimating and planning.

V**ALUABLE** A user story must provide business value, which means it must be valuable for the users or customers of the product.

E**STIMABLE** The Development Team should be able to estimate any user story. This is needed for planning, but it is also a good way of verifying understanding of the story and whether it has the right level of detail.

S**MALL** Or **SIZED APPROPRIATELY**. A user story that will soon be pulled into a Sprint should be small, so that several user stories can be pulled into 1 Sprint. Epic stories can and should be bigger, their size depending on their importance, urgency and risk.

T**ESTABLE** A user story is only good when everybody involved understands how to test and accepts it. Typically, acceptance criteria are added to the story to define and agree on the criteria of success.

User story pitfalls

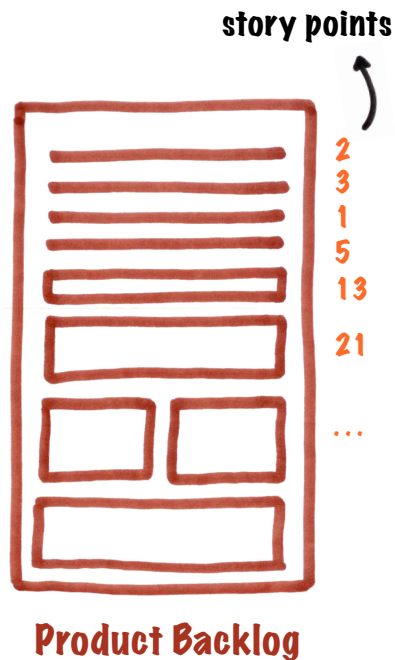
Next to not matching the INVEST criteria, these are some typical pitfalls when writing user stories:

- ✓ **TOO DETAILED OR SOLUTION FOCUSED** The user story contains too much technical or functional detail and/or is describing a solution instead of a business problem, need or goal.
- ✓ **USER NOT WELL-DEFINED** The user is defined too generic, no use of persona's, or the user story is defined from a 'system' or technical point of view.
- ✓ **BUSINESS GOAL NOT CLEAR** The business need/goal is missing or described too vaguely. Or it is closer to the 'What' than to the 'Why', often hidden as 'I want X so that X'.
- ✓ **NO ACCEPTANCE CRITERIA** No acceptance criteria (conditions of satisfaction) are added to the story as it gets closer to the Sprint Planning.

Agile estimation and planning

Relative estimations

Although not part of the core definition of Scrum, a lot of Scrum Teams use relative estimations for estimating the cost of Product Backlog items. This means:



✓ Estimates are made (only) by the entire Development Team, taking everybody's knowledge into account and increasing commitment as a team.

✓ Estimates are expressed in meaningless, relative (and thus comparable) numbers called (user) story points.

0 1 2 3 5 8 13 21 ...

✓ The Fibonacci sequence (or alike) is used as only possible values for estimates. This tackles the inherent uncertainty in estimating larger items.

✓ The Development Team initially agrees about a small and clear Product Backlog item to use as a first reference (the '1') to start comparing with.

Relative vs. absolute estimates

Why are we not estimating in absolute mandays or hours? Because studies show that relative estimates are:

✓ **made faster**

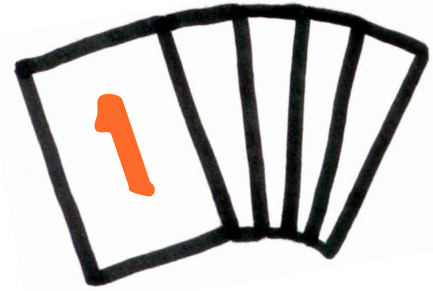
✓ **more accurate**

On top of that, relative estimates are independent of the person that is (later) executing the work. This avoids blaming and hiding and allows the self-organization of the Development Team within the Sprint.

Planning poker

Planning poker is a great technique for making relative estimations. It avoids anchoring and influencing, leading to better estimates.

Next to that, it is a very effective way to implement the Lean principle of **'deciding a the latest responsible moment'** because it helps the team to discuss the Product Backlog items into the right level of detail needed to get decent estimates. This allows the team to get the right questions on the table and take the functional and technical decisions that are needed at that point in time (and no more).



How does it work?

- ✓ The Product Owner briefly explains Product Backlog item.
- ✓ Each team member compares with the reference item (and earlier estimates) and chooses the appropriate card, without showing their card yet.
- ✓ When everybody has chosen, all show their cards together.
- ✓ If needed, people with highest and lowest estimates explain their reasoning, so the team learns about the item.
- ✓ The team gets to a consensus about the estimate, possibly by doing another round of card estimating.

Velocity

$$\text{Velocity} = \text{story points DONE} / \text{Sprint}$$

Velocity is the measurement of how much work a Scrum Team got done in a Sprint. Velocity is expressed in story points, and is the total amount of story points for all the Product Backlog items the team has delivered according to Definition of Done for a given Sprint.

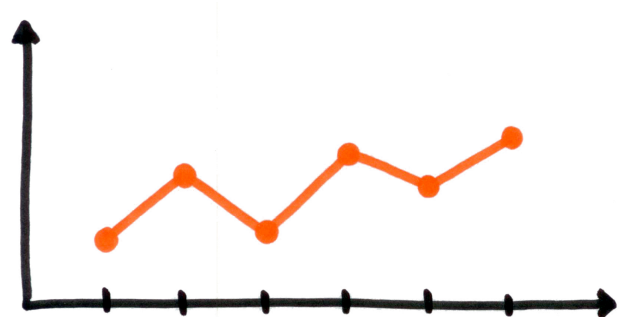
Velocity is used:

- ✓ **By the Development Team:** In the Sprint Planning as one of the criteria to decide how much work to pull into the Sprint. And in the Sprint Retrospective to trigger discussions on increasing effectivity and efficiency.
- ✓ **By the Product Owner:** The team's velocity (eg. average of the last x Sprints) can be used to make forecasts about future releases (release planning).

Velocity always has to be related to Sprint duration and team composition. Velocity is not a metric for management to compare teams or to measure the productivity of a team.

Velocity graph

Scrum Teams visually track their velocity in what is called a velocity graph. This helps to spot trends and discuss things such as the stability of the velocity (predictability) and how to further increase the velocity.



The release burndown graph

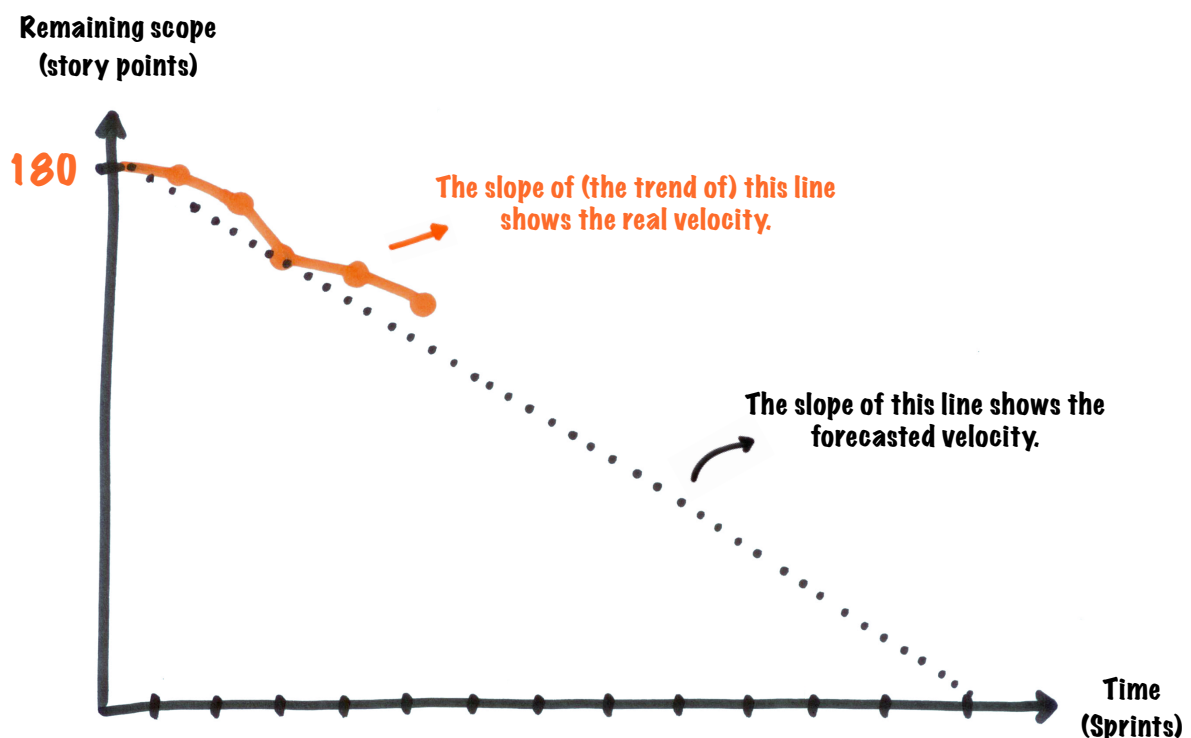
The Product Owner is responsible for release planning: defining and forecasting product releases and the related scope, time and budget. And tracking progress of releases during product development. One of the Product Owner's most important tools is the release burndown graph.

This graph shows the remaining scope (in story points) versus the time and thus visualizes whether the team is on track or not to deliver according to the (initial) plan.



To build this graph, The Product Owner:

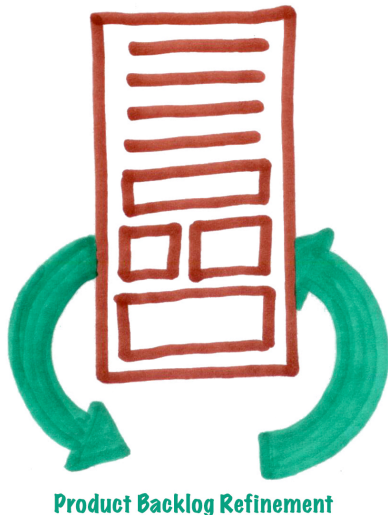
- ✓ Asks the Development Team to estimate all the Product Backlog items that are part of the release.
- ✓ Adds up all estimates to get to the total amount of story points in the release. (180 in our example.)
- ✓ Forecasts the velocity based on measured velocity (eg. average of last 3-4 velocities) or asks the Development Team's initial forecast of velocity (when no data available). And takes into account Sprint duration and team composition.
- ✓ Draws the expected progress, based on the scope, budget and time constraints. (Dotted line in our example.)
- ✓ After every Sprint, measures the remaining scope in the Product Backlog and updates this graph. (Orange line in our example.) Drawing the trendline of the real velocity shows whether the initial plan is feasible or not.



The Scrum Rituals

Product Backlog Refinement

The goal of Product Backlog Refinement is to prepare the Product Backlog for the upcoming Sprint(s). Product Backlog Refinement is not a timebox but an ongoing activity owned by the Product Owner and executed by the entire Scrum Team. It is a collaboration between Product Owner and Development Team. It includes:



- ✓ Adding and removing items to the Product Backlog.
- ✓ Splitting large items (epic stories) into smaller items (epic stories or sprintable user stories).
- ✓ Adding details and acceptance criteria to the Product Backlog items.
- ✓ Ordering the Product Backlog.
- ✓ Estimating the effort/cost of Product Backlog items.

The Product Owner must make sure to do the right Product Backlog Refinement so that the Product Backlog is ready for the next Sprint Planning.

Ordering the Product Backlog

When ordering the Product Backlog, the Product Owner takes the following criteria into account:

- ✓ **CUSTOMER or BUSINESS VALUE** Although hard to measure or quantify, the Product Owner must find ways to estimate the business value of each Product Backlog item. This should drive the ordering of the Product Backlog to reach the long-term vision.
- ✓ **RISK** The Product Owner might put risky items on top of the Product Backlog to mitigate important (technical or business) risks or verify critical assumptions.
- ✓ **COST** Next the value, also the cost of items is important to take into account when ordering to assure maximal return on investment.
- ✓ **DEPENDENCIES** The Product Owner will also need to take (technical and business) dependencies into account when ordering.

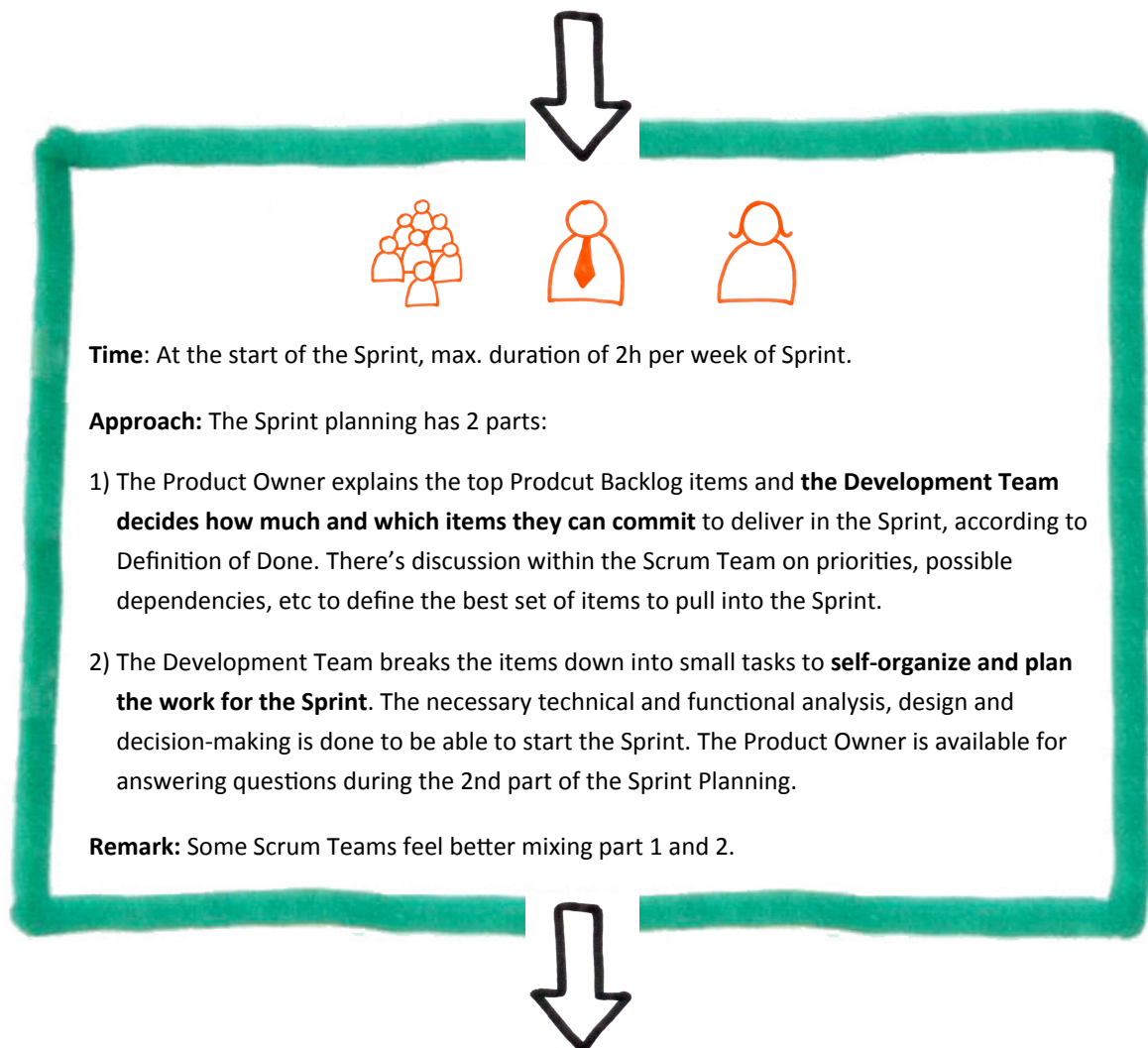
Sprint Planning

The goal of the Sprint Planning is for the Scrum Team to forecast and plan the Sprint. The Development Team members decide how much work is pulled from the Product Backlog into the Sprint Backlog, taking into account the Product Owner's priorities. And they agree on how to get this work done.

A 'ready' Product Backlog, which means:

INPUT

- ✓ It is ordered by the Product Owner based on value, risk, cost and dependencies.
- ✓ The top items are small and 'sprintable' (INVEST).
- ✓ The top items are estimated by the Development Team.



The Sprint Backlog, containing:

OUTPUT

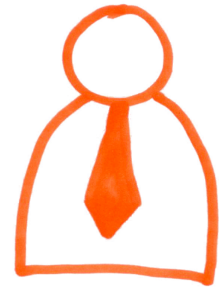
- ✓ The forecast of the team which Product Backlog items will be delivered in this Sprint.
- ✓ A work breakdown of these items and the plan for accomplishing the work.

The Product Owner in the Sprint Planning

The most important thing for the Product Owner related to Sprint Planning is to be prepared. He must have done the right amount of Product Backlog Refinement together with the Development Team so that the Product Backlog is ready for Sprint Planning.

In the Sprint Planning, the Product Owner:

- ✓ Explains the Product Backlog items and his decisions on the order/priority.
- ✓ Answers the team's functional and business questions about the items.
- ✓ Agrees with the team on possible changes in the order of Product Backlog items, to take into account dependencies, team staffing and skills, etc to allow the Development Team to maximize the delivered value in the Sprint.
- ✓ Refines the functional acceptance criteria of Product Backlog items together with the Development Team if needed



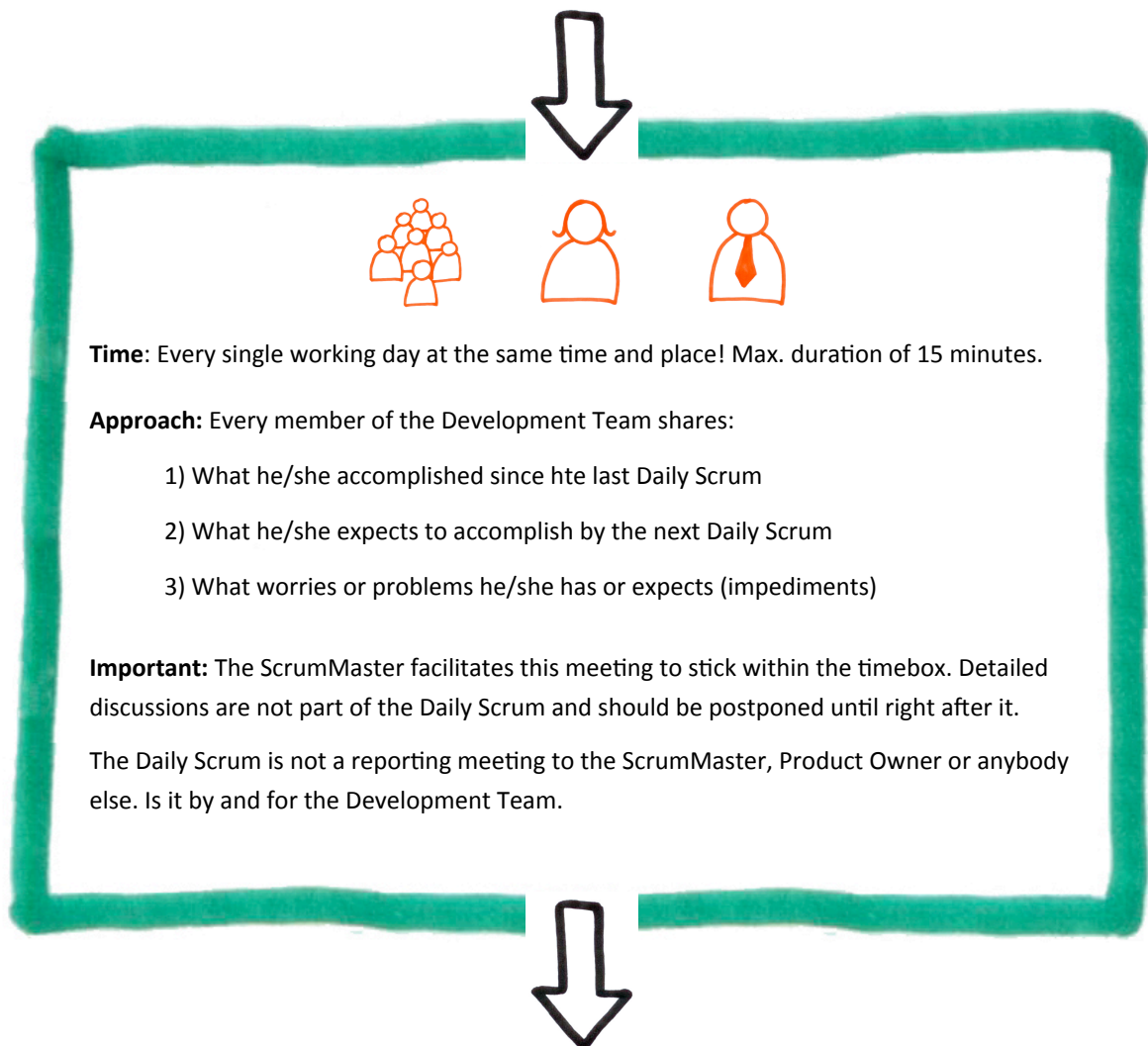
Often, the Product Owner takes the lead in the first part of Sprint Planning, explaining the top items in the Product Backlog and supporting the Development Team in deciding on how much work to pull into the Sprint. When the Development Team starts breaking down and planning the work, the Product Owner usually takes a step back. But remains close and available to the team to answer any questions they might have.

Daily Scrum

The goal of the Daily Scrum is for the Development Team to transparently discuss the progress of the Sprint, possible impediments and what is needed to reach the Sprint goal.

INPUT

- ✓ The experiences and progress of the last 24 (working) hours. Often no physical input (documents etc.) is needed, although team members do come prepared to the Daily Scrum.



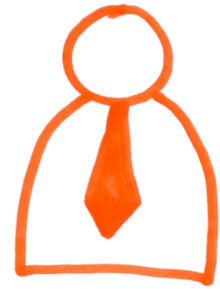
OUTPUT

- ✓ Transparency on who is doing what and about the progress (and impediments) towards the Sprint goal. The Sprint Backlog is brought up to date.
- ✓ Agreement within the Development Team on how to tackle the next working day.

The Product Owner in the Daily Scrum

The Product Owner is not mandatory In the Daily Scrum. But better Product Owners attend the Daily Scrum regularly to:

- ✓ Check in with the team to feel how they are doing.
- ✓ Find out whether the Development Team has questions he should answer or whether he can help removing impediments.
- ✓ Briefly explain the team the work he is doing (related to the product) and agree on upcoming Product Backlog Refinement sessions or actions.

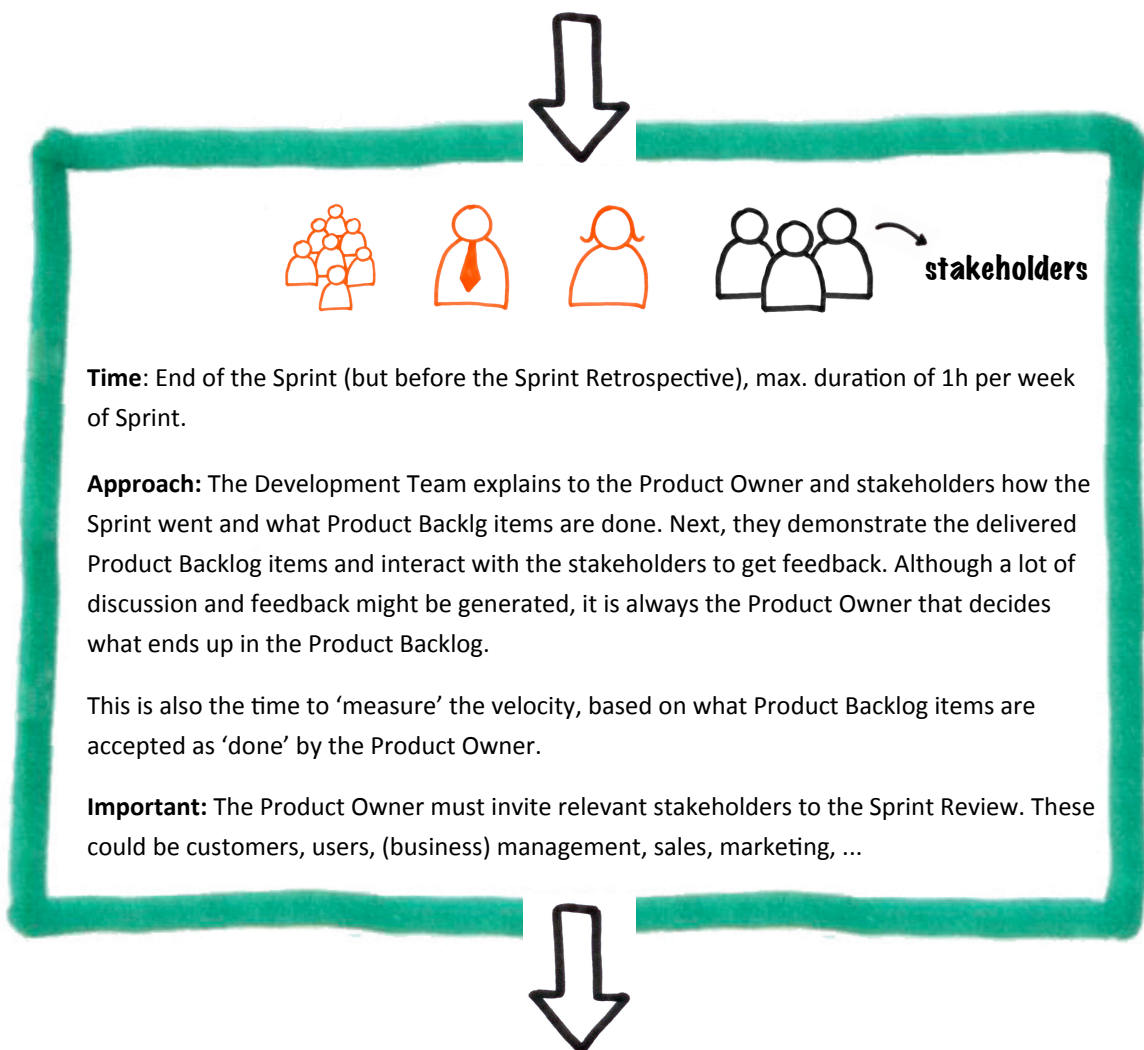


Sprint Review

The goal of the Sprint Review is for the Scrum Team and interested stakeholders to meet and to review the delivered Product Increment. This meeting is about validating and learning about the delivered product and gaining insight in how to proceed (on product level).

INPUT

- ✓ The Product Increment that got delivered during this Sprint.
- ✓ Clarity about whether all forecasted Product Backlog items are truly done (according to Definition of Done) at the end of the Sprint.



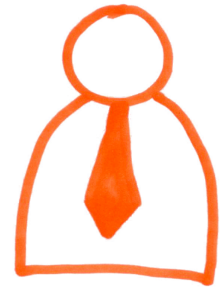
OUTPUT

- ✓ Most important: the done Product Increment!
- ✓ Buy-in and feedback on the product from the stakeholders. Resulting in new ideas for the Product Owner for future Product Backlog items.
- ✓ Insight in how customers engage with the product, leading to better implementation decisions by the Development Team.
- ✓ The velocity of this Sprint.

The Product Owner in the Sprint Review

In the Sprint Review, the Product Owner:

- ✓ Invites and brings relevant (business) stakeholders: customers, users, sponsor, marketing, sales, ...
- ✓ Formally accepts / rejects the Product Increment based on his acceptance criteria and the Definition of Done.
- ✓ Explains his product vision and decisions to stakeholders if needed.
- ✓ Gathers the product feedback of stakeholders and Development Team to update the Product Backlog with if needed.



Often, the Product Owner takes the lead in this meeting, facilitating the content of the discussions. And giving credit to the team for delivering the done Product Increment.

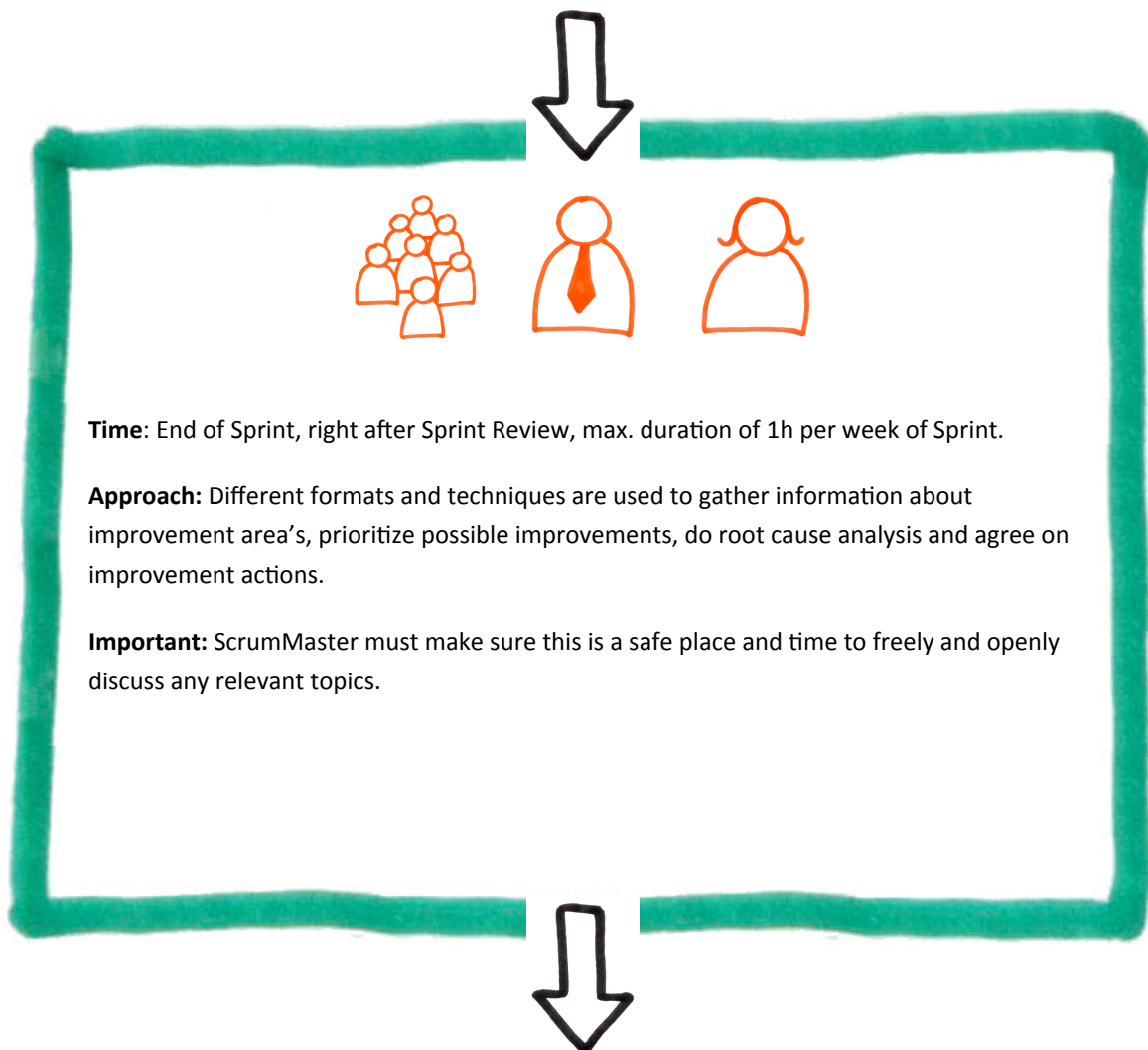
A good Product Owner collaborates with the Development Team during the Sprint to deliver maximal value in the Sprint Review. The Product Owner is prepared for the Sprint Review and knows the results of testing the Product Increment and thus what items to accept and reject.

Sprint Retrospective

The goal of the Sprint Retrospective is to reflect on the past Sprint, considering process, collaboration, tools, productivity and happiness. And to build an action plan for improving things in the upcoming Sprint(s).

INPUT

- ✓ Experience and knowledge gathered in the previous Sprint(s)
- ✓ Feedback gathered in the Sprint Review
- ✓ Forecasted and realized velocity of previous Sprints
- ✓ Previous improvement plans and their current status



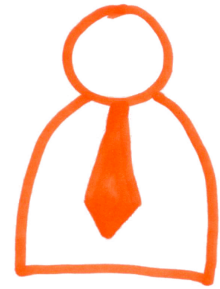
OUTPUT

- ✓ Realistic, actionable and agreed improvement plan. Contains actions and agreements that are executable in the upcoming Sprint(s).

The Product Owner in the Sprint Retrospective

In the Sprint Retrospective, the Product Owner participates in the same way as the Development Team members do. All participants are equal.

If the relationship with the Product Owner is bad, this is sometimes reflected by the Development Team or ScrumMaster not wanting the Product Owner to participate in the Sprint Retrospective. It is indeed more important to have a safe environment in the Retrospective than having the Product Owner participate. But the ScrumMaster should in that case consider the bad relationship to be an important impediment and work on that.



If the team wants to dedicate a Sprint Retrospective to eg. a technical issue in which the Product Owner cannot help, they might of course agree with the Product Owner not to join the meeting.