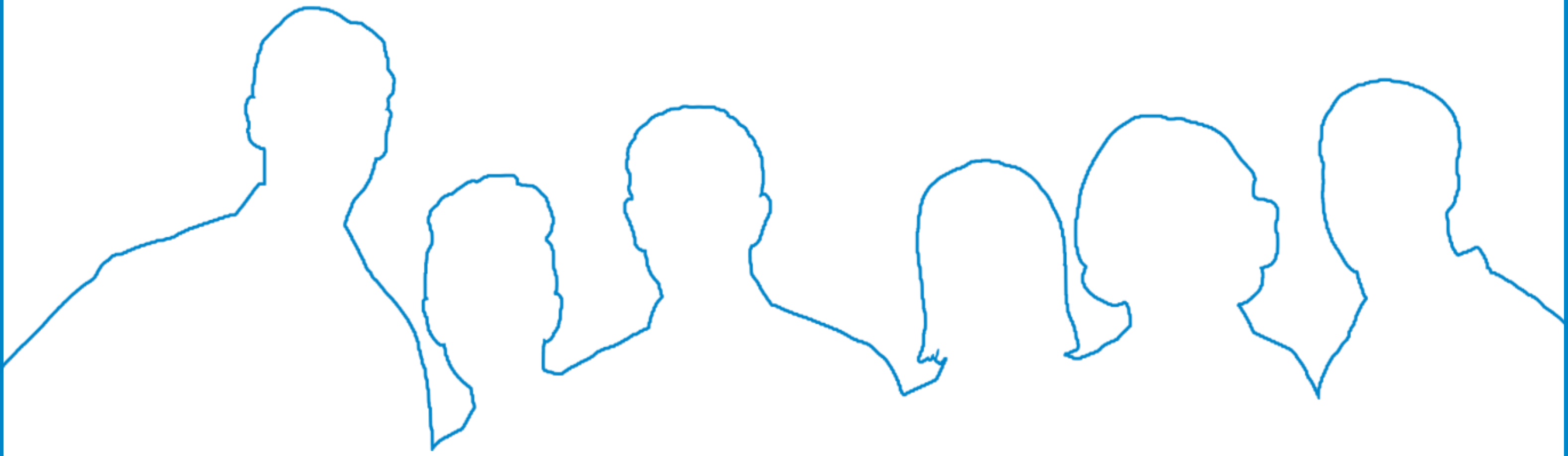


The Scrum Master Training Manual

A Guide to Passing the

Professional Scrum Master™ (PSM) Exam

Version 1.6

By Nader K. Rad, Frank Turley

ISBN 978-90-821147-1-3

9 789082 114713

MANAGEMENT PLAZA
THE MANAGEMENT CERTIFICATION COMPANY

The Scrum Master Training Manual

A Guide to Passing the Professional Scrum Master™ (PSM) Exam

By Nader K. Rad, Frank Turley
Management Plaza

Vr. 1.6 (check for newer versions at http://mplaza.pm/)

Table of Contents

About the Authors

Nader K. Rad is a project management author and adviser at Management Plaza. His career started in 1997 and he has been involved in many projects in different industries. He has designed a number of project management courses, prepared a number of e-learning materials, and written more than 40 books and plenty of practical articles on project management concepts and standards, planning software, scheduling, etc.

Nader has been the official reviewer for PRINCE Agile™, and the new edition of PRINCE®, a contributor to the PMBOK® Guide, and consultant for the Agile Scrum Master™ certification program.

He is certified as PMP®, PRINCE2® Practitioner, MoP® Practitioner, MSP® Foundation, M_o_R® Foundation, MoV® Foundation, P3O® Foundation, AgilePM® Practitioner, ITIL® Foundation, CSM®, PSM I, PSPO I, EXIN Agile Scrum Foundation, and Advanced Certification in Structured and Critical Thinking.

Author's website: http://mplaza.pm/

Author's LinkedIn Profile: linkedin.com/in/naderkrad

Frank Turley has been a project manager for more than 15 years and a PRINCE2® Practitioner and Scrum Master. He is also a PRINCE2 and Project Management trainer and coach and has written a number of PRINCE2® and Project Management related books. Frank is best known in the PRINCE2 world for his work in creating the most popular PRINCE2 Self Study training materials, including:

- The PRINCE2 Foundation Training Manual and video course
- The PRINCE2 Practitioner Training Manual
- The PRINCE2 Sample Project

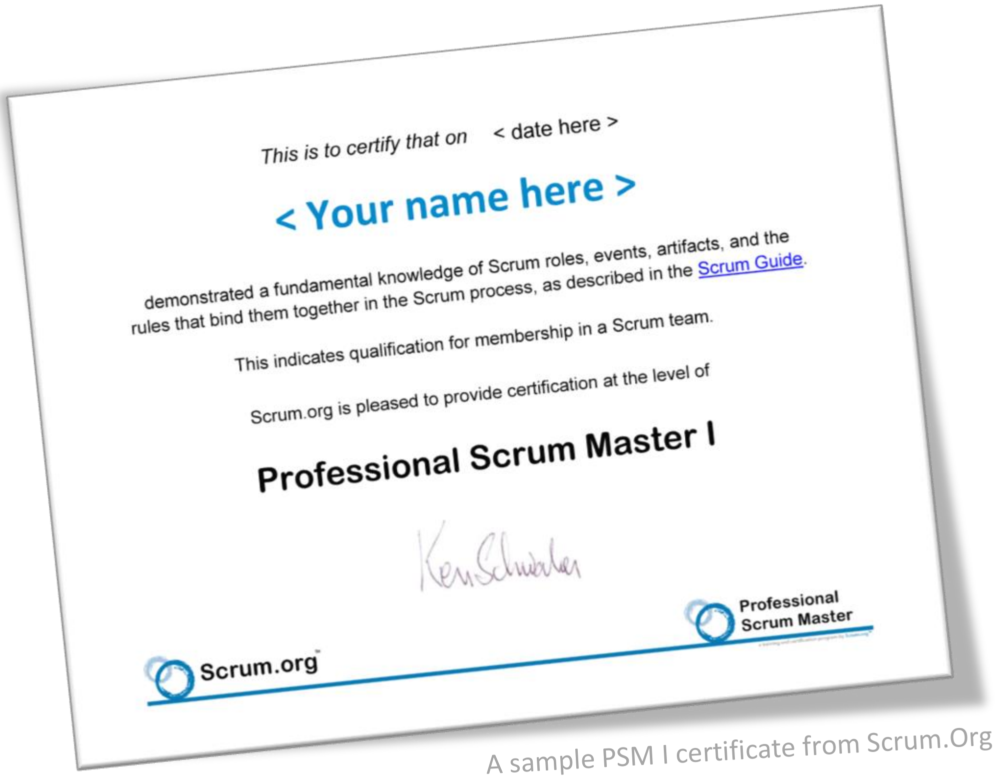
Author's website: http://mplaza.pm/

Author's LinkedIn Profile: http://linkedin.com/in/frankturley

About the PSM™ I Certification

Scrum.org is a leading certification body for Scrum, which was founded in 2009 by Ken Schwaber and Alex Armstrong. Its main certification on Scrum, called PSM™ I (Professional Scrum Master ™ level 1) is one of the most famous certificates in the Scrum community. It validates gained fundamental knowledge of the Scrum framework and its application.

The PSM I Exam is 80 multiple choice, multiple answer, and true/false questions. The duration of the exam is 60 minutes. The passing score is 85%. The exam costs $150 and you can take it online. You need to understand the official Scrum Guide (from Scrum.Org), which you will find very easy to read after reading this book.



This is to certify that on < date here >

< Your name here >

demonstrated a fundamental knowledge of Scrum roles, events, artifacts, and the rules that bind them together in the Scrum process, as described in the Scrum Guide.

This indicates qualification for membership in a Scrum team.

Scrum.org is pleased to provide certification at the level of

Professional Scrum Master I

Ken Schwaber

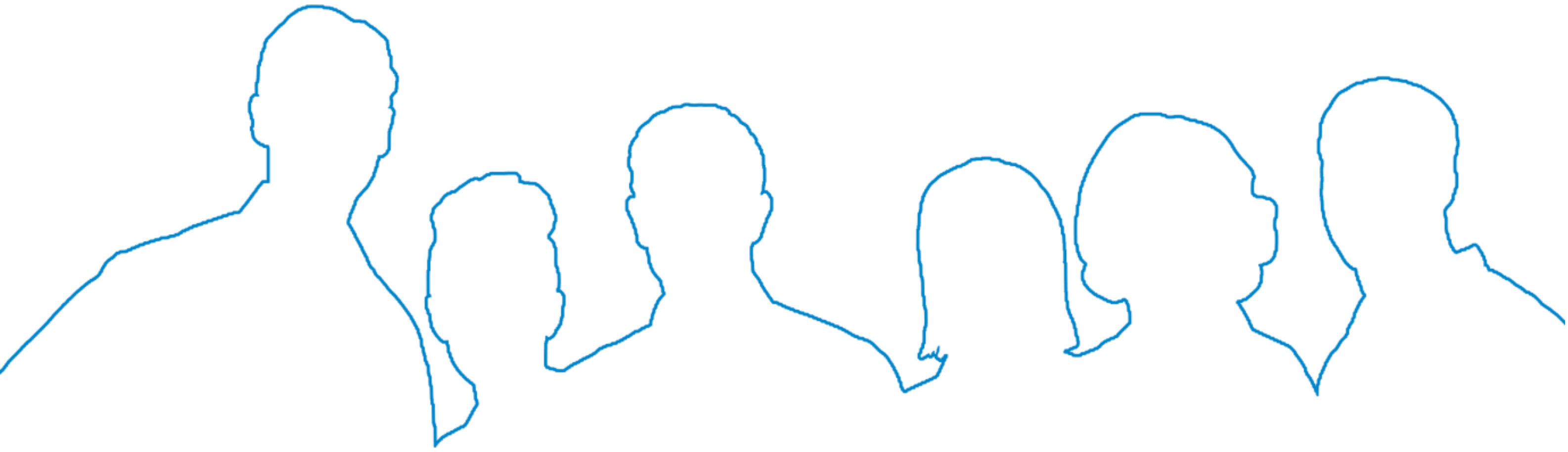
Scrum.org

Professional Scrum Master

A sample PSM I certificate from Scrum.Org

This book helps you understand the Scrum Guide better, and easier. We also provide the following two materials for the exam. Samples of both are available in mplaza.pm.

Introduction



1 Introduction

1.1 What Is Scrum and Agile?

It is not possible in some projects to gather all the requirements upfront because of their extreme uncertainties. So we use an “adaptive” method instead of a “predictive” one, in which we only define and plan a short period in the future, create a working product, and use the product to understand the needs and use the feedback to plan further and add more functionalities to the product.

Adaptive methods are called Agile. There are many Agile frameworks. The most famous Agile framework is Scrum.

Agility is about project development, but they also explain a little about the project management aspects in such environments. However, remember that the main point of Agile frameworks is not providing a project management system, but a project development system.

1.2 When do We Need to Use Agile?

Agility is helpful when it’s hard to define the product upfront, because we can’t understand it entirely, or the customer keeps changing its mind. In those cases, if you try to use a predictive method with upfront plans, you will have many change requests, and it will decrease your productivity and increase time and cost.

1.3 When Can We Use Agile?

Agility is based on adaptation, which in turn is based on the capability of producing pieces of working product throughout the project and using that for receiving feedback from the end users or their representatives. The interim outputs should be working product, because otherwise we won’t be able to receive real feedback.

The working product is always “incremental”, which means we keep adding new features to it. The development should be “iterative”, which means we should repeat the development processes (e.g. design) for each Increment.

We can only use Agile when the product has the capability to be developed iteratively and incrementally. Unfortunately, not every product has it. The best case for Agility is software development. It’s also used in some research projects, and many programs (e.g. organizational change initiatives).

1.4 Agile Manifesto

In 2001, a group of software developers published a manifesto that has since been considered the core of all Agile methods.

The complete Agile manifesto is as follows:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

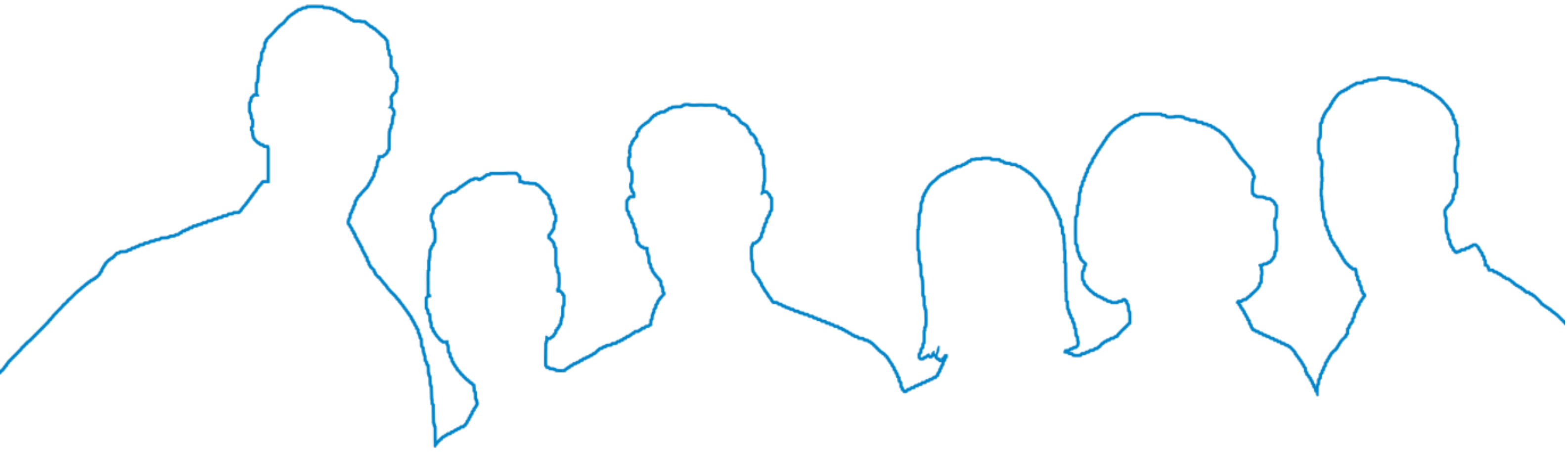
Individuals and interactions	Over	processes and tools
Working software	Over	comprehensive documentation
Customer collaboration	Over	contract negotiation
Responding to change	Over	following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Scrum is a simple way of realizing all these values.

The creators of Scrum, and other Agile frameworks/methodologies such as XP (eXtreme Programming), Crystal, and DSDM were among the people who composed the manifesto.

Scrum Roles

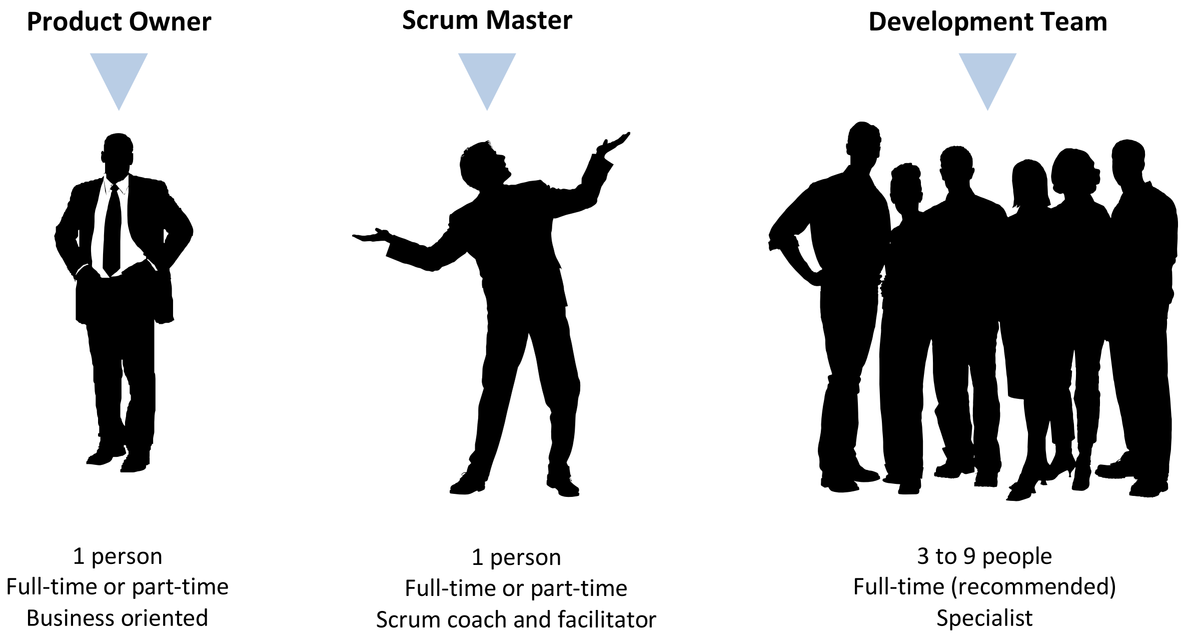


2 Scrum Roles

2.1 Scrum Team

There are three roles in a Scrum project; no less, and no more. We are not allowed to define any other roles, because it is harmful to the unity of the team, and it is not compatible with the philosophy of Scrum.

A *Scrum Team* consists of the following three roles:



Product Owner

1 person
Full-time or part-time
Business oriented

Scrum Master

1 person
Full-time or part-time
Scrum coach and facilitator

Development Team

3 to 9 people
Full-time (recommended)
Specialist

The term “Scrum Team” refers to all the project team members: everyone internal to the project. Scrum Team members usually have only one of the three standard roles of Scrum: Product Owner, Scrum Master, or Development Team member. It is possible for a single person to be assigned to more than one of the standard roles, but it is not recommended.

Other persons can also be involved in the project but they are not considered *internal* to the project and Scrum theory does not have much to say about them. They should have a certain set of behaviors though, to make it possible for a Scrum project to succeed.

The customer should understand and adopt the Scrum framework too, as the relation between the customer and the performing organization, and the way we deliver the project completely changes when we switch to the Scrum framework.

Note: in case of internal projects, the term “customer” will refer to another part of the company that orders the product and will probably use it in their operation afterward.

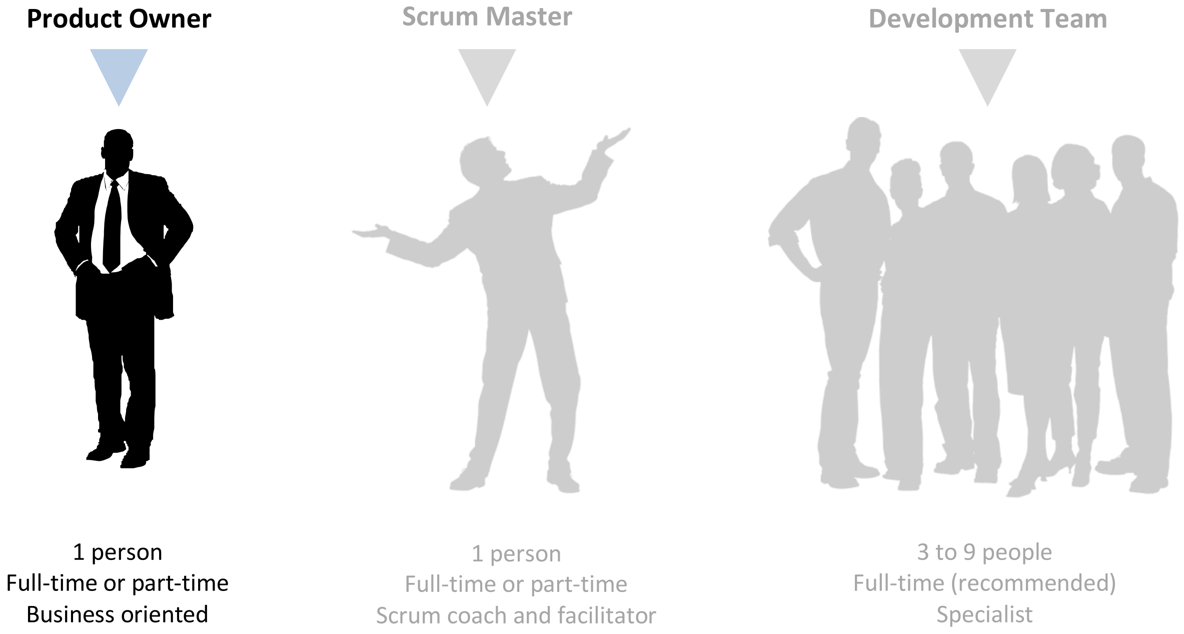
The Scrum Team has two essential characteristics:

- **Self-organized**: The Scrum Team manages its own efforts rather than being managed or directed by others. In traditional methods, management efforts are separated and centralized; a subset of the project team is responsible for project management and others are only responsible for specialist activities. However, management and specialist efforts are *not* separated in Scrum.
- **Cross-functional**: The Scrum Team has all the expertise and competencies needed to get the job done without any help from outside the team.

These two characteristics are designed to optimize flexibility, creativity, and productivity, needed for the Agile environment of Scrum.

It might be required to have more team members for larger projects. In that case, we can use multiple teams for a single product, and it is called scaled Scrum. Scaled Scrum should follow the whole Scrum framework nevertheless.

2.2 Role 1: The Product Owner



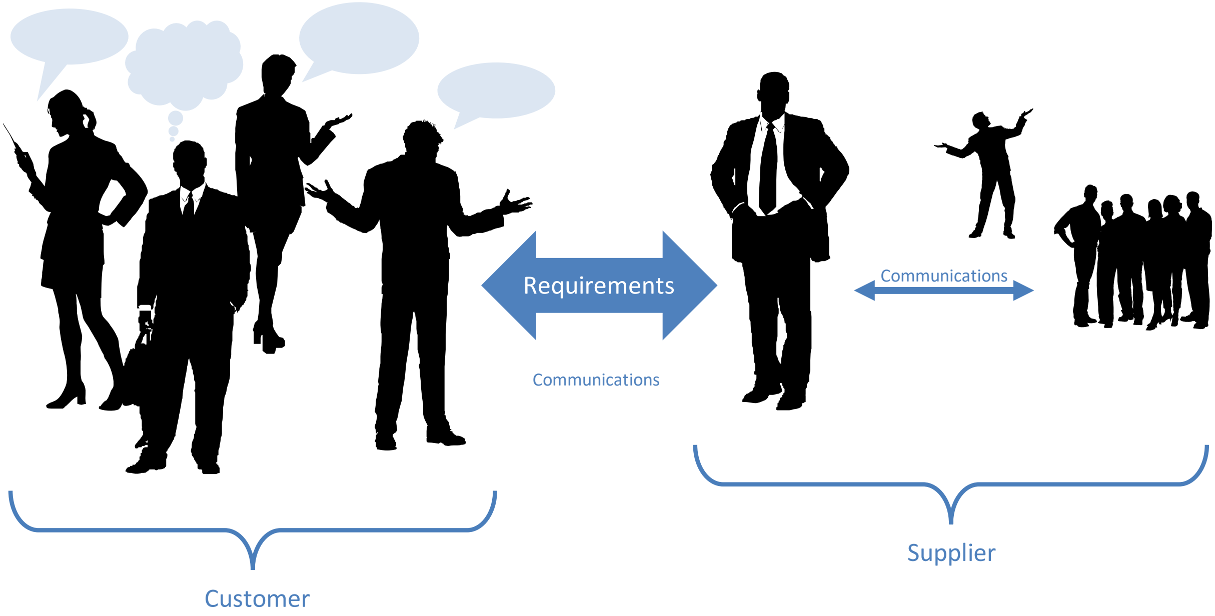
Each project needs a business oriented person, aimed at maximizing the value of the product and the work of the Development Team. In Scrum, this person is called Product Owner. Product Owners are normally a person from the supplier company, rather than from an external customer. In other words, Product Owner is NOT the customer's representative.

This role belongs to one person. There can be a committee to handle the responsibilities of this role, but in such a case, there should be one person representing this committee and we call this one person the Product Owner. There's only one Product Owner, even if you are using scaled Scrum with multiple teams.

They do not need to have application area knowledge of the project; they are focused on the business aspect. In software development projects for example, Product Owners do not need to be developers themselves; they just need to know a little about development, but a lot about how the business operates.

The Product Owner is responsible for the *Product Backlog*. The Product Backlog is a prioritized list of items (usually user stories) that the customer expects from the project; this is the main planning tool in Scrum. It is also the responsibility of the Product Owner to make sure that each Product Backlog item is easy to understand for the Scrum Team, and other stakeholders.

Product Owners should communicate effectively with the customer (the inevitable success factor in every project), and use the information to keep the Product Backlog updated with all the changes. They also measure the performance of the project, forecast the completion date, and make this information transparent to all stakeholders.



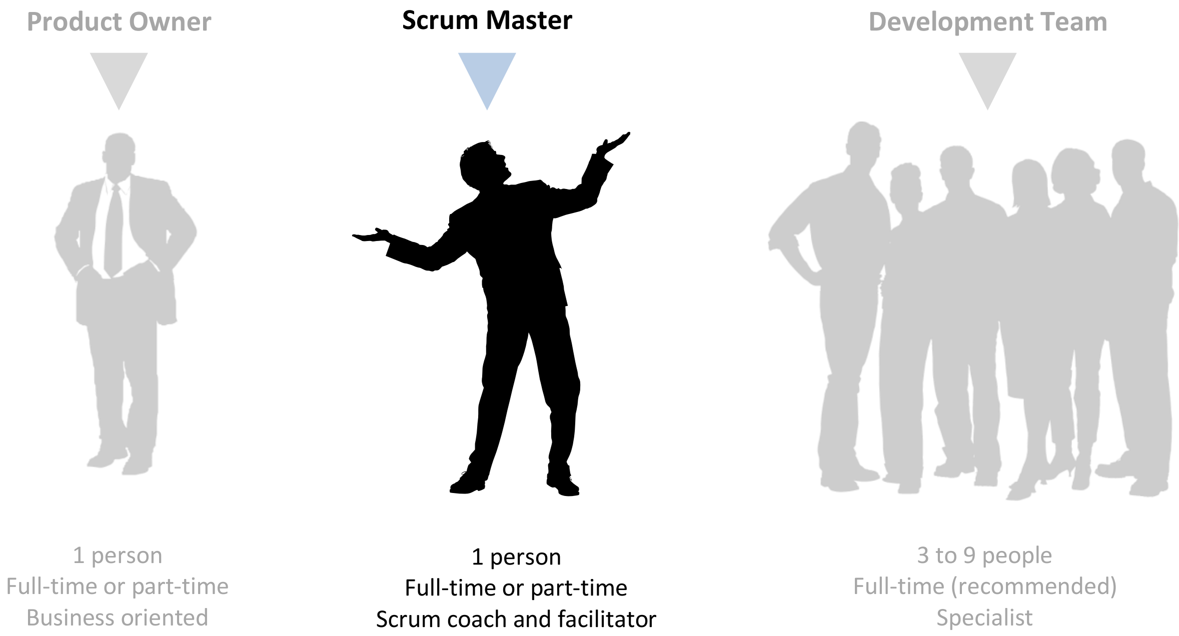
Product Owners understand the business, so they can rank each Product Backlog item based on its return on investment, as well as any other factors they find suitable for the business point of view of the project. The items will be sorted based on their *value*, so the higher they are on the list, the sooner they will be developed by the Development Team.

The entire organization must respect the Product Owner decisions for the project to be successful. No one should allow themselves to try to override those decisions, and no one should tell the Development Team what item to deliver, except for the Product Owner. A Product Owner's decisions might be influenced by others, but s/he must have the final say.

A Product Owner might delegate some of her/his responsibilities (such as preparing the list of items for the Product Backlog) to the Development Team, but stays accountable for them.

Remember, there's only one Product Owner, even in scaled Scrum; because it is very hard to manage value otherwise.

2.3 Role 2: The Scrum Master



Product Owner
1 person
Full-time or part-time
Business oriented

Scrum Master
1 person
Full-time or part-time
Scrum coach and facilitator

Development Team
3 to 9 people
Full-time (recommended)
Specialist

Scrum Masters are those who fully understand Scrum, and help the Scrum Team by coaching them, and ensuring that all Scrum processes are implemented correctly. The Scrum Master is a management position, which manages the Scrum process, rather than the Scrum Team. S/he is a servant-leader for the Scrum Team.

Besides ensuring that the Development Team understands and uses Scrum correctly, the Scrum Master also tries to remove impediments to the Development Team, facilitates their events, and trains and coaches them.

The Scrum Masters help the Product Owners too, by helping or consulting them on finding techniques, communicating information, and facilitating related events.

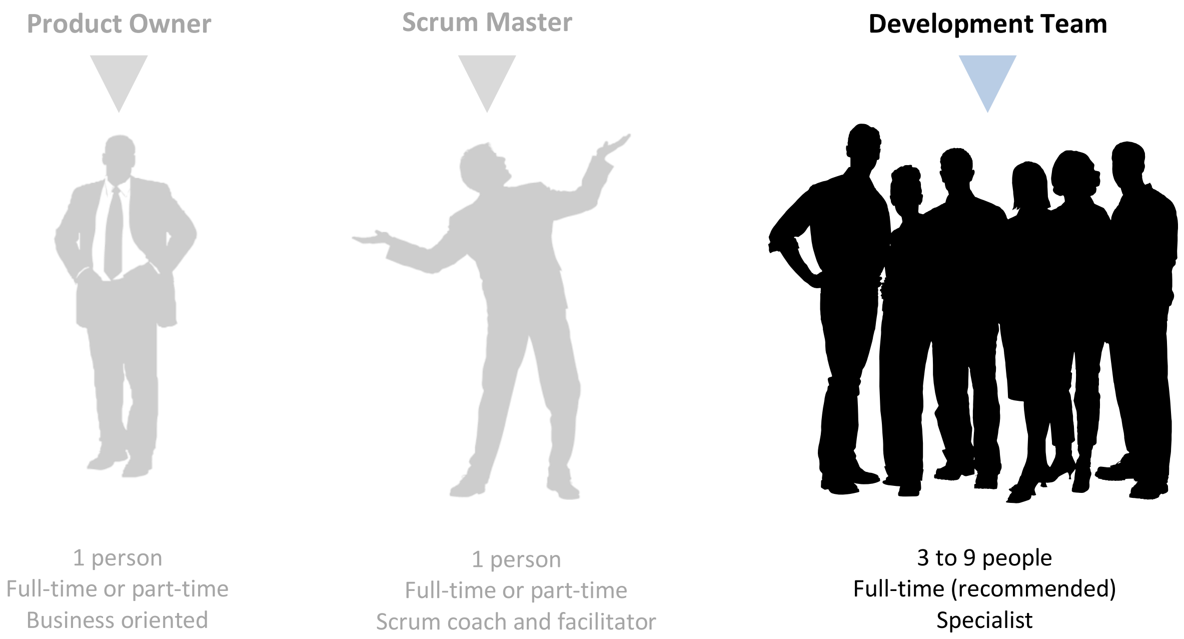
The responsibilities of the Scrum Masters are not limited to the Scrum Team. They should also help those outside the Scrum Team understand the appropriate interactions with the

Scrum Team to maximize the value created by the Scrum Team. The Scrum Master usually leads the organization in its effort to adopt Scrum.

It is possible for a single person to be both Scrum Master, and a member of the Development Team, although this is not recommended. Being a Scrum Master of a project might not occupy 100% of the time of a person; in this case, the best solution is to assign that same person as the Scrum Master in more than one project, rather than making them a member of the Development Team.

There's one Scrum Master role per team in scaled Scrum. However, a single person can be the Scrum Master of more than one team.

2.4 Role 3: The Development Team



Product Owner

1 person
Full-time or part-time
Business oriented

Scrum Master

1 person
Full-time or part-time
Scrum coach and facilitator

Development Team

3 to 9 people
Full-time (recommended)
Specialist

Members of the Development Team are application area experts that are responsible for delivering backlog items, *and* managing their own efforts.

They should be cross-functional; being capable of doing the A to Z of the creation of each Product Backlog item. They should be self-organized: find their own way instead of receiving orders. They should be aligned with the goal of the project instead of working blindly. A task might be assigned to a single member throughout the Sprint, but the whole Development Team will be responsible and accountable for that task; no individual owns any task.

The Development Team delivers the final product of the project in step by step Increments, as defined in the Product Backlog. They always work in a product-based way.

It is highly recommended for members of the Development Team to work full-time on a single project, to stay focused and agile. The composition of the Development Team should not change so often. If there is a need to change team members, then this change should not happen during a Sprint. There will be a short-term decrease in productivity when the composition of the team changes.

Scrum is mostly effective when there are 3 to 9 Development Team members. For large projects, we can use a scaled model with multiple Scrum Teams.

2.5 Other Roles and Titles

You might have the temptation to give Development Team members more specific titles, such as designer, tester, quality inspector, and team leader; but Scrum does not allow this. All members should have the same role, and the same title: Development Team member.

Scrum is completely depended on collaboration and team-work. Development Team members should be united and completely aligned with the goal of the project. If you give them different titles or roles, they will focus on their own specific role in the project instead, and they might not pay enough attention to the final product which is necessary for Agile projects. Each Development Team member is responsible for all the outputs created in the Development Team, even though each of them might be focused on a specific set of tasks.

2.6 So Who Is the Project Manager?

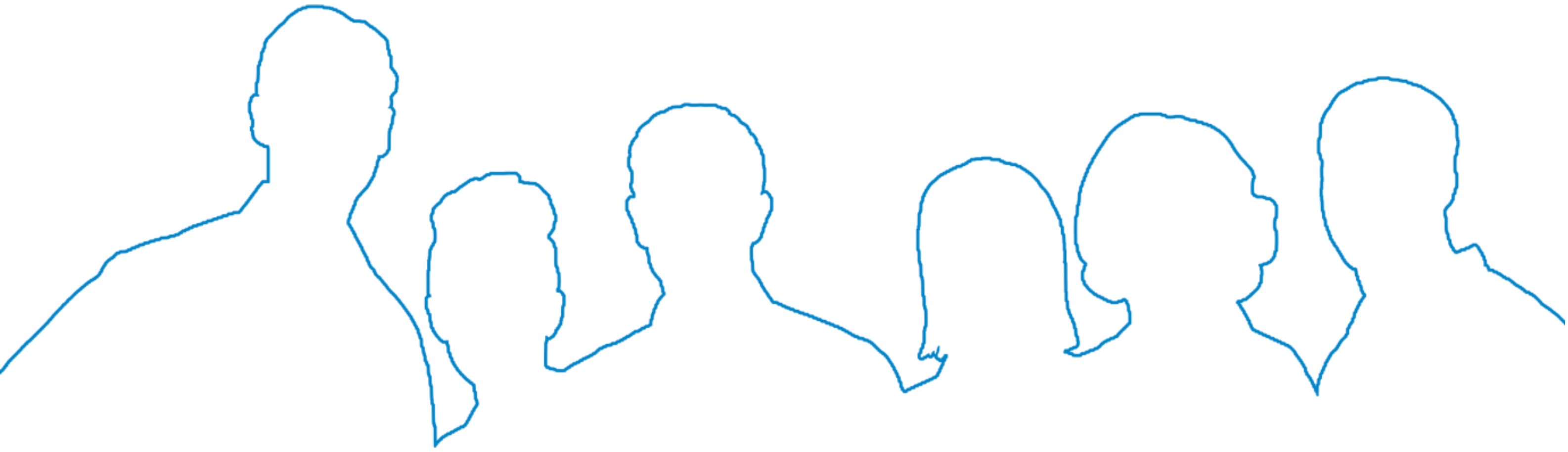
Now that we have reviewed all the Scrum roles, you might ask yourself, who is the *project manager*?

The answer is simple: there is no such role in Scrum; and none of the 3 roles of Scrum acts as a traditional project manager.

The project management responsibilities are *distributed* among the three roles of Scrum and there is no centralized project management in Scrum.

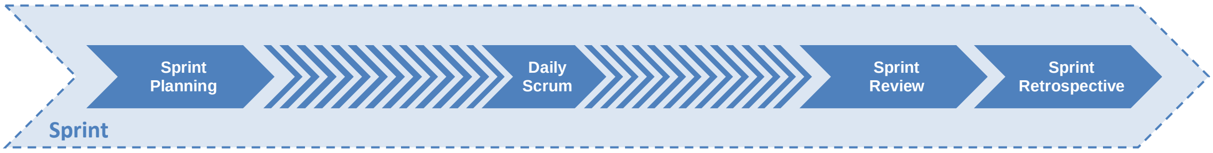
> Note that having a project manager is not against Agility; it’s just not applicable to Scrum. Other Agile methods, such as DSDM, have project managers.
>
> You can start a free email course on DSDM here, if you’re interested.

Scrum Events



3 Scrum Events

3.1 The Nature of Scrum Events



There are five events in a Scrum project:

1. **Sprint**: Each Scrum project is a set of Sprints. A Sprint is a container for the four other events (as represented in the above diagram), development effort, and the maintenance of the Product Backlog.
2. **Sprint Planning**: Sprint Planning is the first event inside a Sprint. The Scrum Team plans the items they are going to deliver in the Sprint and the way they will deliver them.
3. **Daily Scrum**: The Development Team starts working on the objectives of the Sprint as soon as Sprint Planning is completed. During the Sprint, the Development Team holds a daily meeting (normally 15 minutes) to coordinate the work for the next 24 hours. This meeting is called the Daily Scrum.
4. **Sprint Review**: Before the end of the Sprint, the Development Team reviews the outcome of the Sprint with the customer to receive feedback. The feedback is used to adjust the Product Backlog.
5. **Sprint Retrospective**: After the Sprint Review and just before the Sprint is over, the Scrum Team holds an internal meeting to review the Sprint (lessons learned) and use it to improve the process in the next Sprint. This meeting is called Sprint Retrospective.

The four events inside the Sprint are designed to enable critical transparency, inspection, regularity, and adaptation. We prefer to use these predefined meetings with fixed objectives and time-boxes (maximum durations) instead of ad-hoc meetings, which have the potential to waste our time.

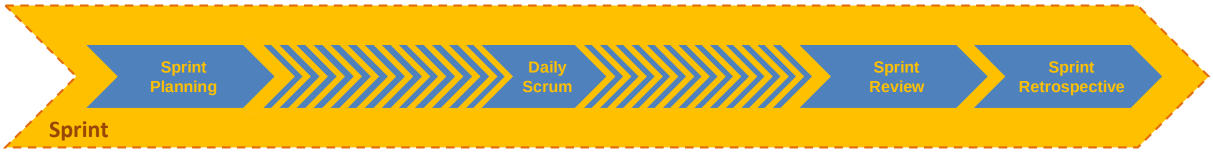
There is an essential concept in Agile methods, called time-box: a predefined maximum duration of time. In order to maximize productivity, all the Scrum events must be time-boxed. It helps everyone focus on the real problems, instead of going into too much unnecessary detail.

3.2 The Time-Box Concept

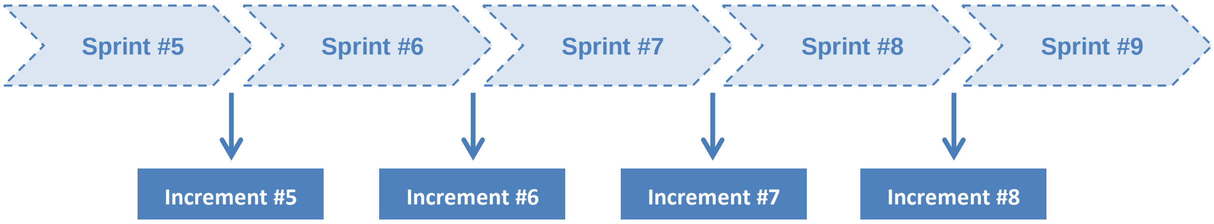
Time-box is an essential concept in Scrum. It is our way of staying focused and getting things done in an ever-changing environment. A time-box has a maximum duration in which we freeze the target and work on certain tasks or objectives.

The duration of a time-box Sprint should be agreed upon and fixed. We are free to change the duration based on lessons learned, but not frequently, and never based on single occasions. For example, we are not allowed to say that “we have a lot to do this time, so let’s increase the duration for this particular case”. What we are allowed to say is “based on the previous ten Sprints, we’ve realized that the duration of our Sprints is unsuitable, and a 30% increase in duration might better fit our needs. So, let’s increase it from now on”.

3.3 Event 1: The Sprint



Each Scrum project delivers the product in a number of iterations, which are called Sprints in Scrum. An Increment is developed in each Sprint. An Increment is a potentially releasable product. An Increment is a sum of all Product Backlog items completed so far in a project and this Increment keeps getting bigger after each Sprint. You can think of it as different versions of a software; each time with more features.
Increments may or may not be actually released (put into use), but should always be potentially releasable.



Customers usually request changes when they see the Increment (during the Sprint Review), and we note these new requests in the Product Backlog.

Sprint is a time-boxed event, which means we should fix its duration at the beginning of the project and not change it frequently or occasionally. Sprints are fixed for one month or less.

An important point is that we do not change the items of the Sprint Backlog after the Sprint is started and the plans are set. The Sprint Goal (discussed further in Sprint Planning) should not change either. The Product Owner and the Development Team might try to clarify and re-negotiate the scope as more is learned about the items, by changing the tasks, but will not *change* the Sprint Backlog items and the Sprint Goal. Even the composition of the Development Team and the quality expectations should not change during a Sprint. These constraints are designed to make it possible to focus and get things done.

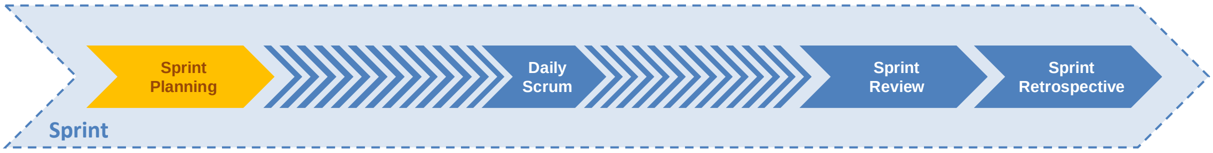
Each item in the Product Backlog should normally be completed in a single Sprint as this is much easier to manage. The Development Team selects a number of items from the top of the Product Backlog (this has already been prioritized by the Product Owner) and aim to get them "Done" (100% complete) during the Sprint. We want them to be really "Done" when the Sprint is over, and create an Increment. An Increment is the sum of all the completed items during a Sprint and all previous Sprints. However, it's OK if they are not able to deliver all items. If they are pressured to deliver everything, they will start selecting fewer items to be safe, and it will consequently lower their productivity because of the "student syndrome". The student syndrome says that work expands to fill the available time.

It is important to agree on a definition of "Done" at the beginning of the project. We will not call something "Done", unless it fits the definition. A 99.999% completed item is not considered "Done", it would not be part of the Increment and it would not be demonstrated to the customer at the Sprint Review; it will be returned to the Product Backlog and if it's still at the top (after reprioritization), it will be selected for the next Sprint.

Sprint Time-boxes: Most companies use Sprint time-boxes of 2 to 4 weeks. If we use Sprints longer than one calendar month, it will be likely for the unapplied changes to become large enough to create problems. This will increase the complexity and risk. Sprints should not be too short either, because we would not be able to produce complete Backlog items during it. Our goal is to deliver the final product item by item, inside the Sprints; we do not want to split a single Product Backlog item among several Sprints.

Can a Sprint be canceled? Even though Sprint Backlog items do not change, the Product Owner has the authority to *cancel* a Sprint. This can happen when the Sprint Goal becomes obsolete, due to changes in the Product Backlog, strategies, approach, market, etc. When a Sprint is canceled, the items that are "Done" will be reviewed and accepted, and the rest of the items (not started or partly complete) will be put back into the Product Backlog to be done in the future.

3.4 Event 2: Sprint Planning



The Development Team does not wait until the Product Backlog is 100% planned (all requirements are gathered and cleared) to start developing the project. As soon as the Product Backlog is mature enough (has the necessary number of stories) which will provide the information for the Sprint, the Scrum Team can start the first Sprint.

The first thing to do in each Sprint is Sprint Planning. Sprint Planning is a time-boxed meeting, usually fixed to 8 hours for a one month Sprint, or shorter for Sprints of less than a month. All three roles should attend this meeting.

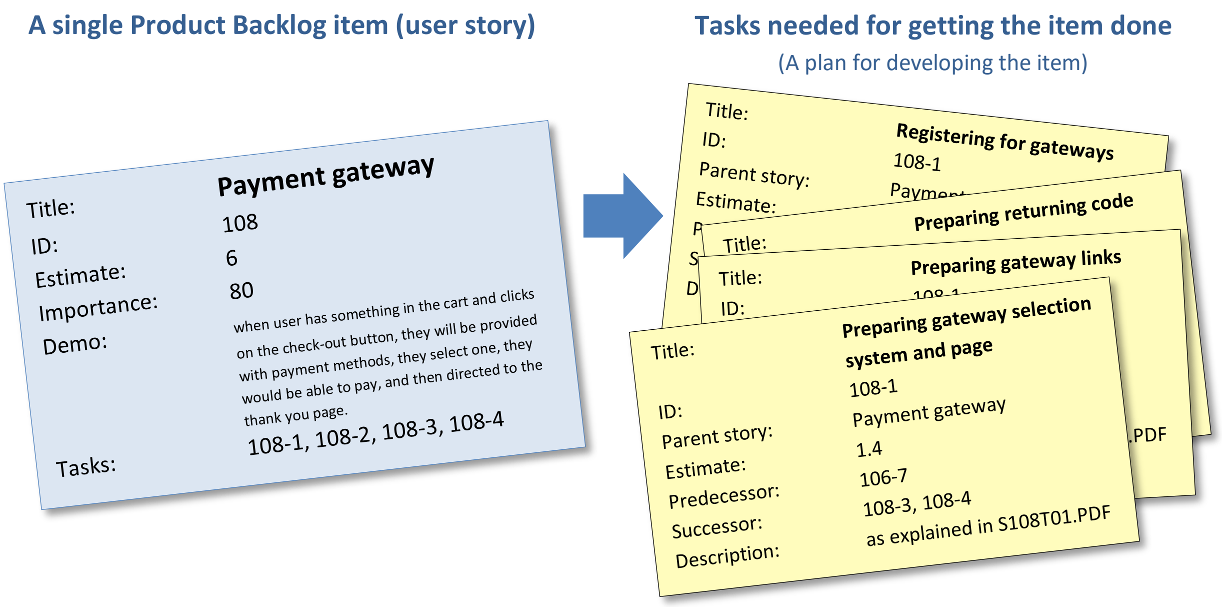
The Development Team estimates the capacity of work it can deliver in a single Sprint. The Product Owner has already ranked and ordered the Product Backlog based on the business value of the items. The Product Owner also ensures that the items are easy to understand. The Development Team then selects an appropriate number of items from the top of the Product Backlog, and puts them in the Sprint Backlog, to deliver in the current Sprint. The amount of work for each item is estimated by the Development Team and the total amount of work of the selected Product Backlog items is close to the estimated capacity of the Development Team.

The Scrum Team composes a Sprint Goal. The Sprint Goal is an objective that should be met within the Sprint through the implementation of the Product Backlog. The Scrum Goal provides guidance to the Development Team on why it is building the Increment.

The scope of the Sprint, which is made up of the items selected from the Product Backlog, needs more details through the Sprint. These details should be aligned with the Sprint Goal, and likely re-negotiations for them should be done in presence of the Product Owner.

When the items are selected and the Sprint Goal is agreed, it is time to plan how they will deliver the items into a “Done” product Increment and realize the Sprint Goal. This is the second element of the Sprint Backlog: tasks. Not all tasks are planned in this event; having a detailed plan for the first few days is enough. The Development Team can prepare detailed plans for the rest of the work later on. After all, Agility is about replacing upfront planning with adaptation.

A detail plan, is a breakdown of a Product Backlog item into detailed tasks needed to be done in order to create the item. Each task might have estimates, dependencies, and similar information to make tracking possible.



A single Product Backlog item (user story)

Title: Payment gateway
ID: 108
Estimate: 6
Importance: 80
Demo: when user has something in the cart and clicks on the check-out button, they will be provided with payment methods, they select one, they would be able to pay, and then directed to the thank you page.
Tasks: 108-1, 108-2, 108-3, 108-4

Tasks needed for getting the item done
(A plan for developing the item)

Title: Registering for gateways
ID: 108-1
Parent story: Payment
Estimate:

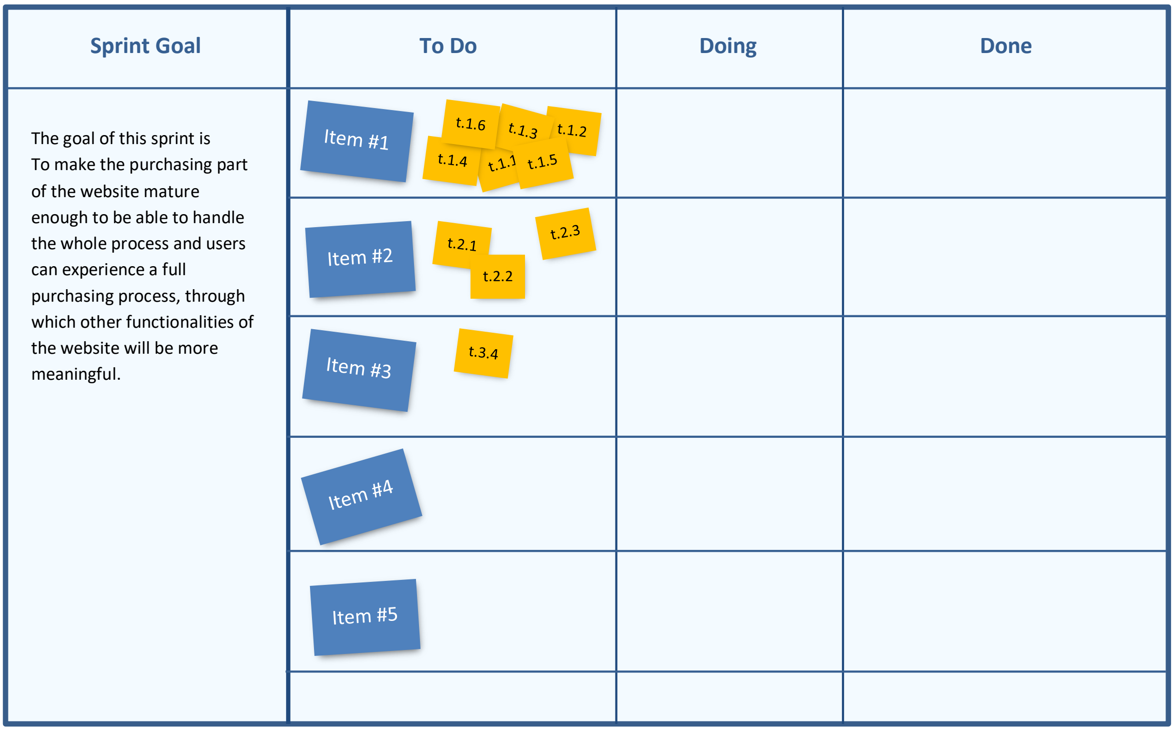
Title: Preparing returning code

Title: Preparing gateway links
ID: 108-1

Title: Preparing gateway selection system and page
ID: 108-1
Parent story: Payment gateway
Estimate: 1.4
Predecessor: 106-7
Successor: 108-3, 108-4
Description: as explained in S108T01.PDF

The Sprint Backlog will be ready at the end of this meeting and the Development Team should be able to describe what items they will deliver through the Sprint, and how they will do it.

There is no specific rule on documenting, storing, and presenting the Sprint Backlog. It can be on a board similar to one shown in the next figure.



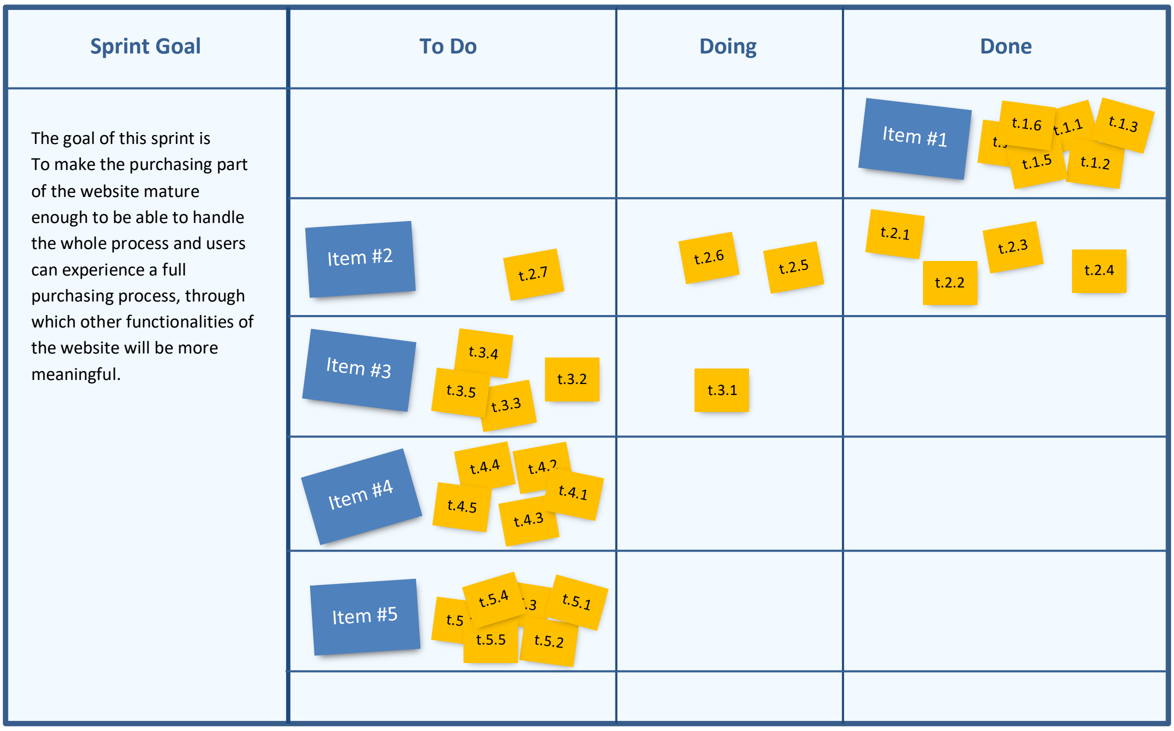
Sprint Goal	To Do	Doing	Done
The goal of this sprint is To make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.	Item #1 t.1.6 t.1.3 t.1.2 t.1.4 t.1.1 t.1.5		
	Item #2 t.2.1 t.2.3 t.2.2		
	Item #3 t.3.4		
	Item #4		
	Item #5		

Yellow sticky notes on the above board are tasks that are created by breaking down items. These tasks define what the Development Team will do to deliver each item, and they are responsible for preparing them. Some tasks are created at the Sprint Planning meeting, and some others throughout the Sprint.

The Sprint Backlog consists of the following:

1. Selected items from the Product Backlog, to be delivered through the Sprint (they do not change during the Sprint)
2. A detailed plan for turning the selected items into “Done” Increment of the product and to realize the Sprint Goal (they evolve during the Sprint)

The Sprint Backlog elements are shown on the sample board above. The board can contain other elements such as the Sprint Goal, and a burn-down chart. This sample Scrum board has extra features for tracking the tasks and items in “To Do”, “Doing”, and “Done” columns as well. The next figure shows the same Sprint after the first item is complete and items #2 and #3 are in-progress.



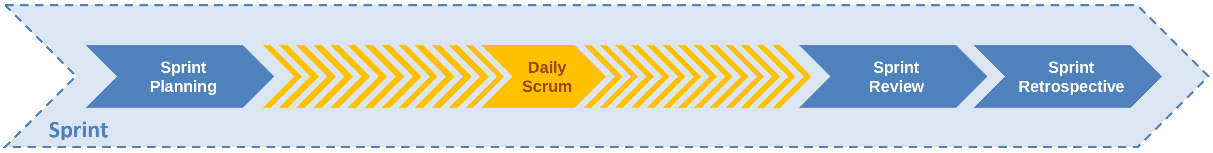
Sprint Goal	To Do	Doing	Done
The goal of this sprint is To make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.			Item #1 t.1.6 t.1.1 t.1.3 t.. t.1.5 t.1.2
	Item #2 t.2.7	t.2.6 t.2.5	t.2.1 t.2.3 t.2.2 t.2.4
	Item #3 t.3.4 t.3.5 t.3.3 t.3.2	t.3.1	
	Item #4 t.4.4 t.4.2 t.4.5 t.4.1 t.4.3		
	Item #5 t.5.4 .3 t.5.1 t.5 t.5.5 t.5.2		

You can also see that extra tasks have been added to the lower priority items (items #3 to #5).

Items in the Sprint Backlog usually have the same order they had in the Product Backlog; therefore, the Development Team should work on the higher items first.

In scaled Scrum, members of all Development Teams will gather and select their items with agreement with the Product Owner. Each team has its own Sprint Backlog.

3.5 Event 3: Daily Scrum



The Daily Scrum is a 15-minute meeting for the Development Team to inspect the work since the last meeting, and synchronize their work and plan for the next 24 hours. It must be held daily.

During the Daily Scrum, each member of the Development Team should answer these three questions:

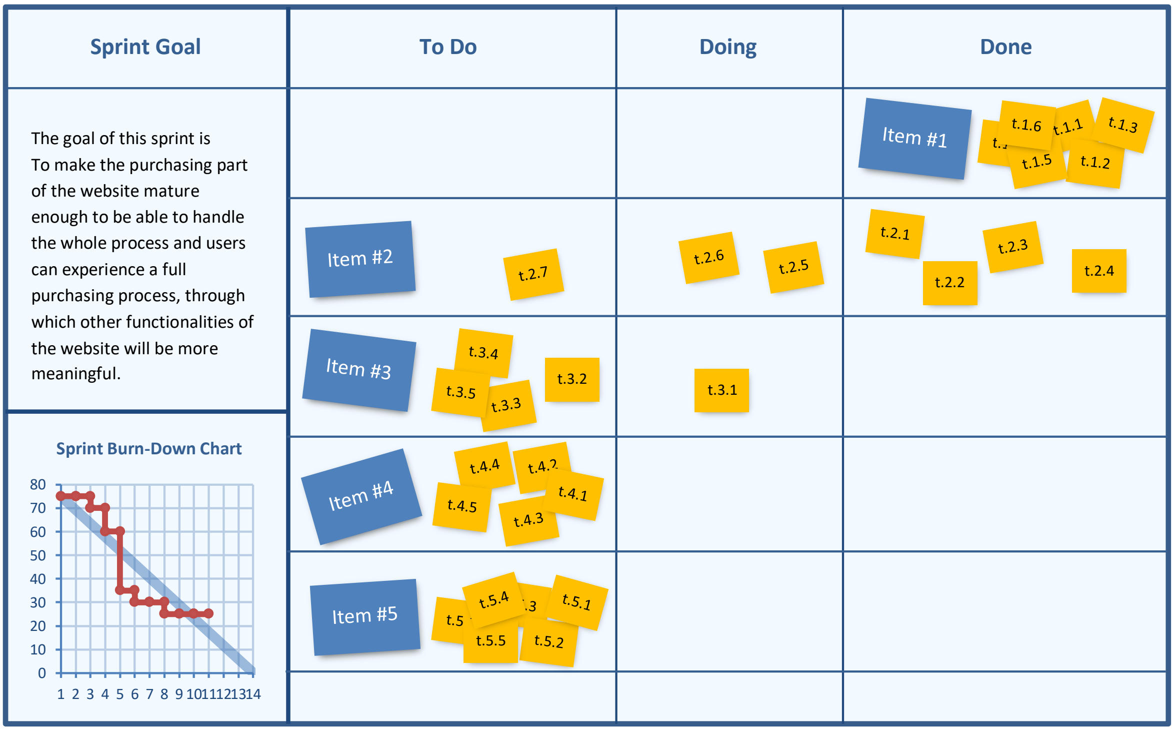
1. What has been accomplished since the last meeting?
2. What will be done before the next meeting?
3. What obstacles are in the way?

Note: obstacles are not discussed in the meeting. If someone has a solution or concern, s/he should mention it after the Daily Scrum.

They assess progress toward the Sprint Goal and forecast the likelihood of completing the items *after* the Daily Scrum. The Daily Scrum is not a status meeting for all the stakeholders; it is just for the Development Team.

The Daily Scrum meeting should be held at the same time and place throughout the Sprint, to minimize complexity.

The Development Team should monitor Sprint progress each day, but not during the Daily Scrum. They can use a burn-down chart to track their remaining work and check to see if they are going to complete all items before the end of the Sprint. The information gathered through Daily Scrums would be useful for updating the board, but the updating is not done during the Daily Scrum.



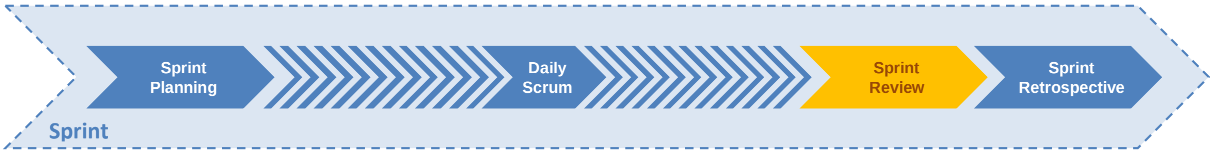
Sprint Goal

The goal of this sprint is To make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.

The above figure contains the Sprint Burn-Down Chart (the tracking information) and this can be updated after each Daily Scrum meeting. Burn-Down Charts are discussed further in the next section.

There's a common extra event in scaled Scrum: Scrum of Scrums. When Development Teams are done with their Daily Scrums, each will send a representative to a higher-level daily meeting called Scrum of Scrums. The representatives will synchronize team activities in this meeting. Each representative answers the three standard questions, plus this: what dependencies will your team have with other teams?

3.6 Event 4: Sprint Review



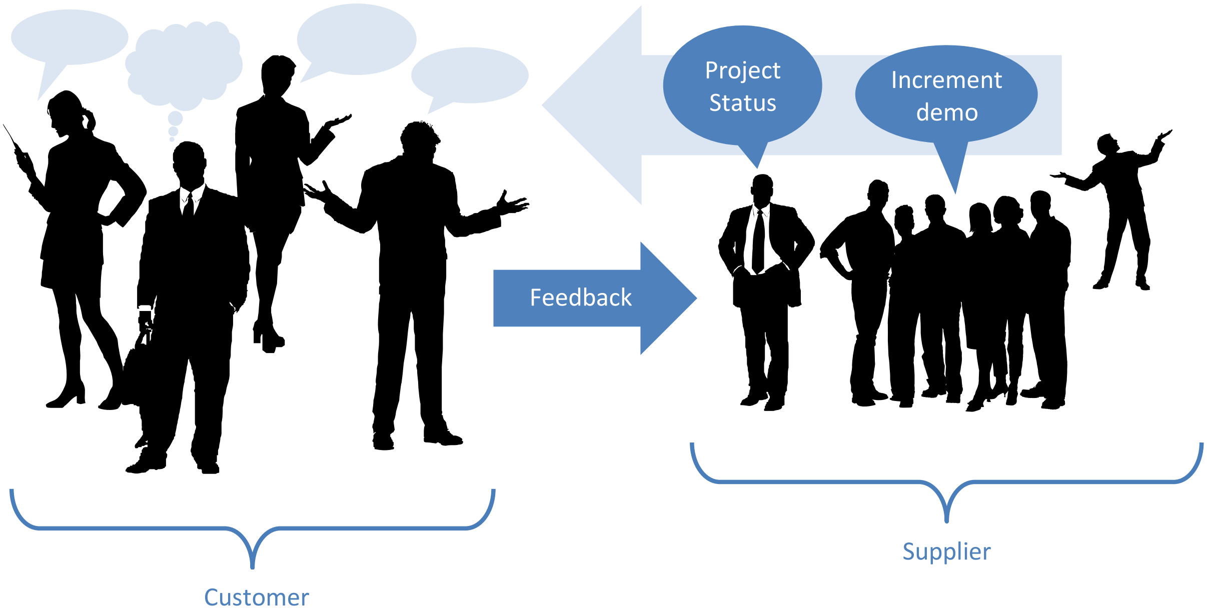
The duration of this meeting is normally **four hours** for a one month Sprint. If the Sprints are shorter then this meeting will be proportionally shorter.

At the end of the Sprint, the Scrum Team and other stakeholders gather and hold a four hour meeting to present the "Done" Increment to the customer. The demonstration is intended to collect feedback and raise change requests at the earliest time possible.

We welcome changes in Scrum and encourage them to be demanded, because it increases the satisfaction of the customer and will create a final product that better matches the needs of the customer. This is the basis for adaptation: we use the Increments to understand the needs and plan further, instead of planning everything upfront.

The Development Team does not present an item, unless it is 100% complete based on the definition of “Done”. The Product Owner makes sure (before the Scrum Review) that presented items are “Done”. The Development Team demonstrates and explains the items.

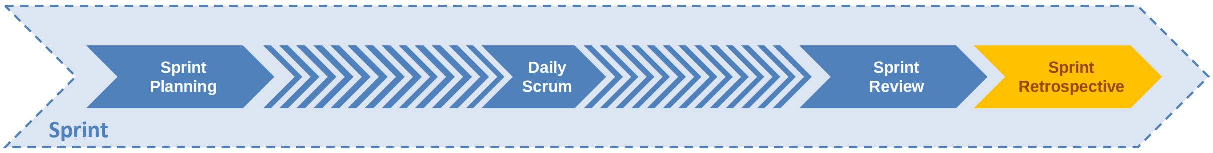
The Product Owner discusses the status of the Product Backlog and the likely completion date based on the progress.



Finally, the whole Scrum Team collaborates on revising the Product Backlog based on the output of the Sprint and the feedback received from the customer.

Remember that Sprint Review is not a formal meeting for receiving approval from the customer; it’s an informal meeting, without exchanging signatures, just aimed at receiving feedback and adjusting the Product Backlog.

3.7 Event 5: Sprint Retrospective



This meeting is normally **three hours** for a one month Sprint. If the Sprint is shorter than one month, this meeting will be proportionally shorter.

After the Sprint Review and just before the end of the Sprint, another meeting will be held, aimed at process improvement (learning lessons), which is called Sprint Retrospective.

There is a rule: we should always look for ways to improve. It does not matter how little the improvement is, there should be an improvement. This meeting is a formal opportunity for improvement, even though we do not limit our improvement to the results of this meeting. We will review (inspect) the Sprint, with regards to people, relationships, processes, and tools, and identify ways of improving them in the next Sprint.

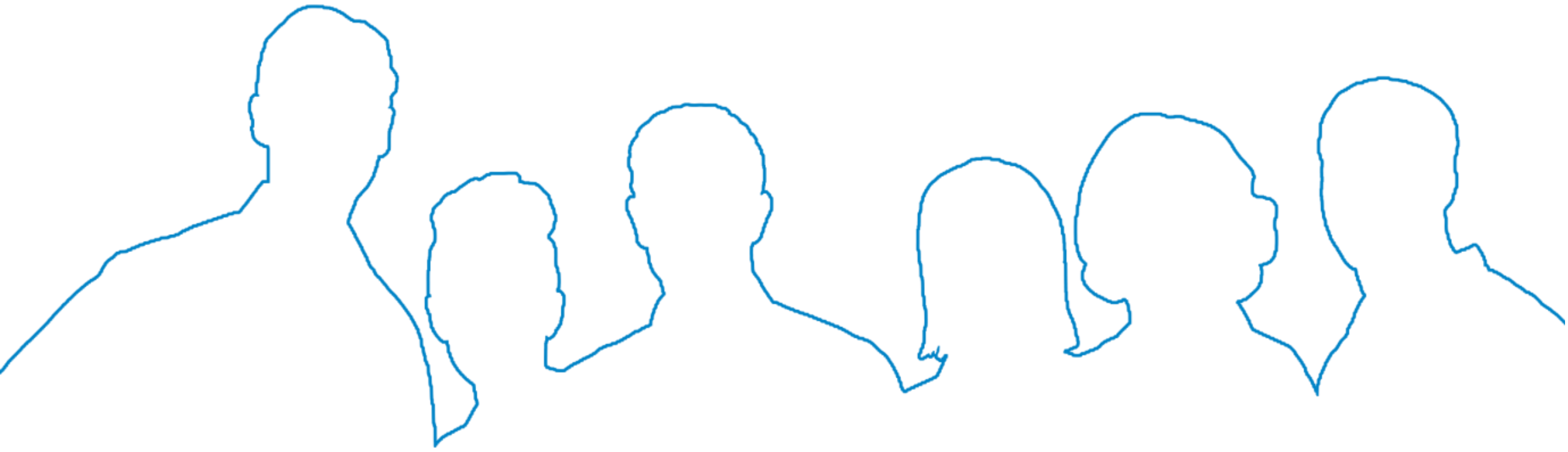
3.8 Activity: Product Backlog Refinement

Besides the time-boxed events discussed before, there is an ongoing activity in Scrum projects called Product Backlog refinement (aka Product Backlog grooming). It is the act of reviewing and revising Product Backlog items, which typically involves adding detail, estimates, and order to them. The Product Owner is responsible for ordering (prioritizing) the items and the Development Team is responsible for estimating.

When the Product Owner adds a new item to the Product Backlog, s/he explains it to the Development Team and asks for estimates. Therefore, everyone knows the meaning of items before the Sprint Planning and the estimates are ready. What happens in the Sprint Planning is that they review those pieces of information, just to make sure that everything is OK.

The main difference between this activity and the five Scrum events is that Scrum events are all time-boxed, while this is an ongoing activity that happens throughout the Sprint. This activity should not consume more than 10% of the time of the Development Team.

Scrum Artifacts



4 Scrum Artifacts

Scrum artifacts are designed to increase transparency of information related to the delivery of the project, and provide opportunities for inspection and adaptation. They are management products useful for the creation of the specialist product of the project.

There are three artifacts in Scrum:

1. **Product Backlog**: An ordered list of everything that might be needed in the final product
2. **Sprint Backlog**: Selected items from the Product Backlog to be delivered through a Sprint, along with the tasks for delivering the items and realizing the Sprint Goal
3. **Increment**: The set of all the Product Backlog items completed so far in the project (up to the end of a certain Sprint)

And these are the extra concepts related to the artifacts:

- **Definition of "Done"**: The shared understanding of what it means for a piece of work to be considered complete
- **Monitoring Progress towards a Goal**: The performance measurement and forecast for the whole project
- **Monitoring Sprint Progress**: The performance measurement and forecasts for a single Sprint
- **Velocity**: a simple measurement for the average amount of work done during each Sprint.

4.1 Artifact 1: Product Backlog

The Product Backlog is an ordered list of everything that might be needed in the final product of the project; in other words, parts of the expected final product (a wish list). All items are described in simple business language (non-technical) and all of them are presentable to every stakeholder. They should also be independent of each other, as we need to order them based on their business value. Every requirement and every change in the project will be reflected in the Product Backlog.

The Product Backlog is dynamically changing and improving; it is never complete. We do not wait until the Product Backlog is complete to start delivering the items; the first Sprint starts as soon as the Product Backlog has a sufficient number of stories.

The Product Owner sets a number of factors to determine the value of each item for the business (customer). Return on investment is usually one of the factors. All these factors will be summarized into business value (importance).

The Product Backlog items are ordered based on their value, in a way that the higher an item is, the sooner it will be delivered by the Development Team. As the items located at the top of the Product Backlog will be delivered sooner, they will also be more detailed and clear compared to the lower items.

Each Product Backlog item has a work estimate. These estimates are solely done by the Development Team, and are used in comparison to the capacity of the Development Team in a single Sprint, to determine the number of items that will be selected for that certain Sprint. Additional information might be added to each item to help the Scrum Team take control.

The Scrum Team should add details, estimates, and order to the Product Backlog items all the way through the project, which is called *Product Backlog refinement*. It should not consume more than 10% of the time of the Development Team.

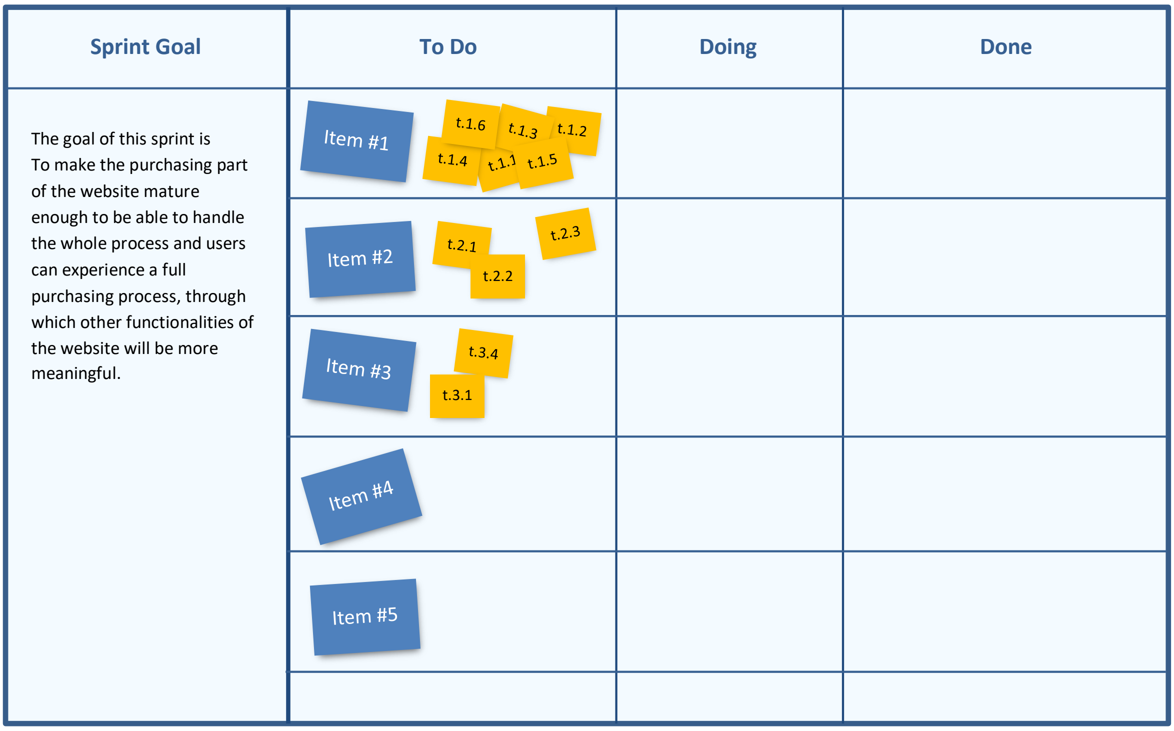
The Product Backlog is created based on *discussion* rather than *documentation*. The Product Backlog items should be easy to understand for non-technical stakeholders.

The Product Backlog is a representation of the scope of the final product and therefore, there should be only one Product Backlog, no matter how many Scrum Teams are working on the project. This single concept also needs a *single* responsible person, so there's only one Product Owner.

4.2 Artifact 2: Sprint Backlog

The Sprint Backlog is created during the Sprint Planning event which is the first event in a Sprint. During the Sprint Planning, the Scrum Team collaborates on creating the Sprint Backlog, which consists of the following:

- A number of items selected from the top of the Product Backlog, based on their estimated work and the estimated capacity of the Development Team
- A detailed plan for delivery of the items and realization of the Sprint Goal during the Sprint (tasks)



Sprint Goal	To Do	Doing	Done
The goal of this sprint is To make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.	Item #1 t.1.6 t.1.3 t.1.2 t.1.4 t.1.1 t.1.5		
	Item #2 t.2.1 t.2.3 t.2.2		
	Item #3 t.3.4 t.3.1		
	Item #4		
	Item #5		

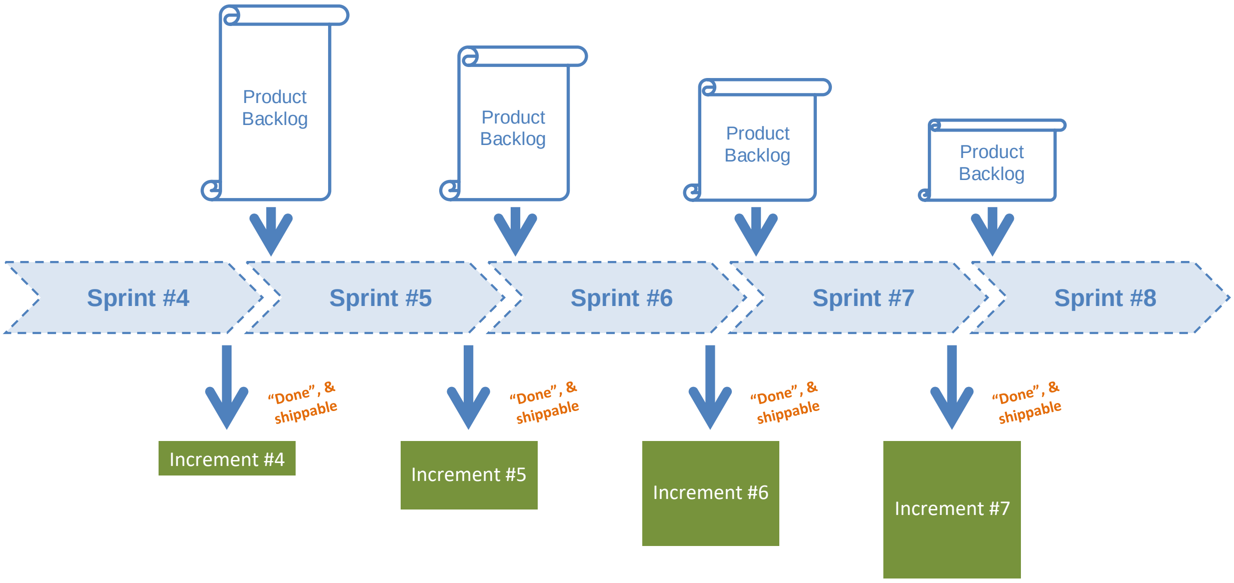
The Sprint Backlog items and the Sprint Goal do not change after the Sprint Planning and the Development Team will focus on delivering an Increment of "Done" based on this plan. However, it is necessary to get more information, justify, or clear some of the items during the Sprint, which should be done in the presence of the Product Owner. The detailed plan (tasks) which is not complete at the end of the Sprint Planning, will continue to be updated as the Sprint continues. So, in general, some parts of the Sprint Backlog do not change during the Sprint, and the others do.

In scaled Scrum, each team needs its own Sprint Backlog.

4.3 Artifact 3: Increment

An Increment is the sum of all completed Product Backlog items at the end of a Sprint. Each Increment must be "Done", which means it should be potentially releasable/shippable and usable by the end user. The Product Owner may or may not release a certain Increment, but it should be releasable nevertheless.

The next figure shows how the number of stories in the Product Backlog decreases Sprint by Sprint, as the number of features in the Increments increases. However, this figure is too simplified, as new items are also added to the Product Backlog throughout the project (especially early in the project).



Note that the Increment concept is cumulative: each Increment also contains the features of the previous ones.

When multiple teams are working on the same product, each team produces a "Done" Increment. The Increments will be combined and create "one" Increment for the project.

4.4 Definition of "Done"

There should be a shared understanding of what it means for a piece of work to be "Done". This definition of "Done" must be discussed and agreed upon by the Scrum Team at the beginning of the project so that future Increments would be releasable.

The definition of "Done" usually contains the following:

- Development processes (e.g. programming, testing, documenting)
- Non-functional features (e.g. security, maintainability, scalability)
- Quality criteria and acceptance criteria

When multiple Scrum Teams are working on a single project, it might not be possible to use the same definition of "Done" for all teams, because they might be working on items of different natures. In such a case, each Development Team defines its own definition of

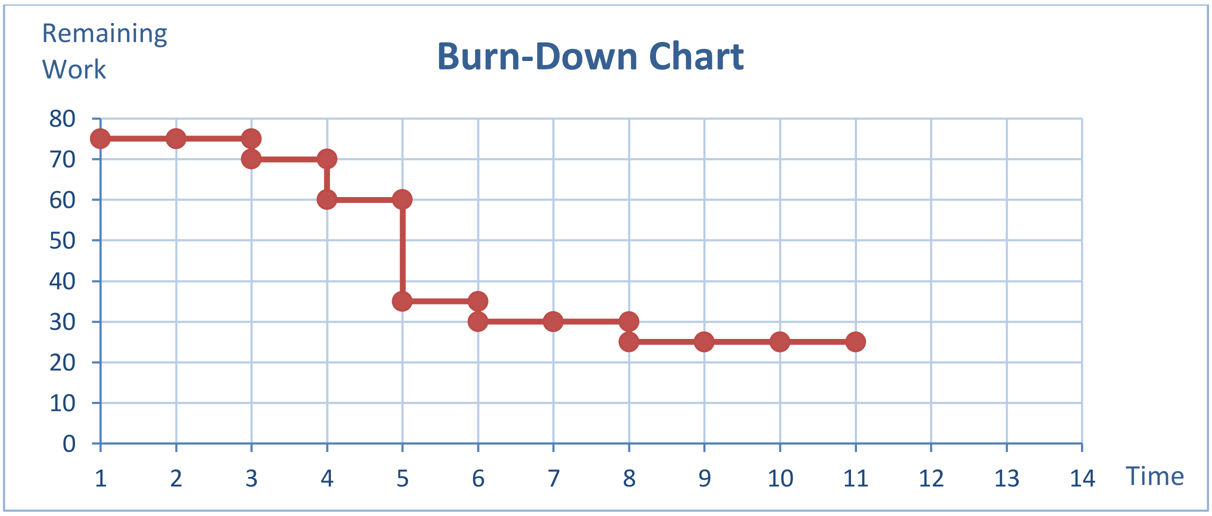
"Done" and delivers its items based on that definition. However, the integration of those definitions of "Done" should be capable of creating a potentially releasable Increment in the project level.

Normally, we expect the definition of "Done" to come from the organization level. This works as the minimum constrains for the project, and then the developers can add more to the definition of "Done", to make it suitable for the project. When there is not a definition of "Done" in the organization, the whole concept is defined by the Development Team.

4.5 Monitoring Progress Toward a Goal

The Product Owner is responsible to monitor the progress of the whole project toward its goal. This should be done at least once per Sprint Review. The Product Owner determines the amount of remaining work and compares it to the remaining work of the previous Sprints, and forecasts the completion date of the project. All stakeholders should have access to this information.

It's common to use a burn-down chart to visualize the progress of the whole project. The project burn-down chart shows the amount of **remaining work**, instead of the amount of completed work; therefore, the line for actual performance goes downward as we proceed and the faster it goes down, the happier we will be!



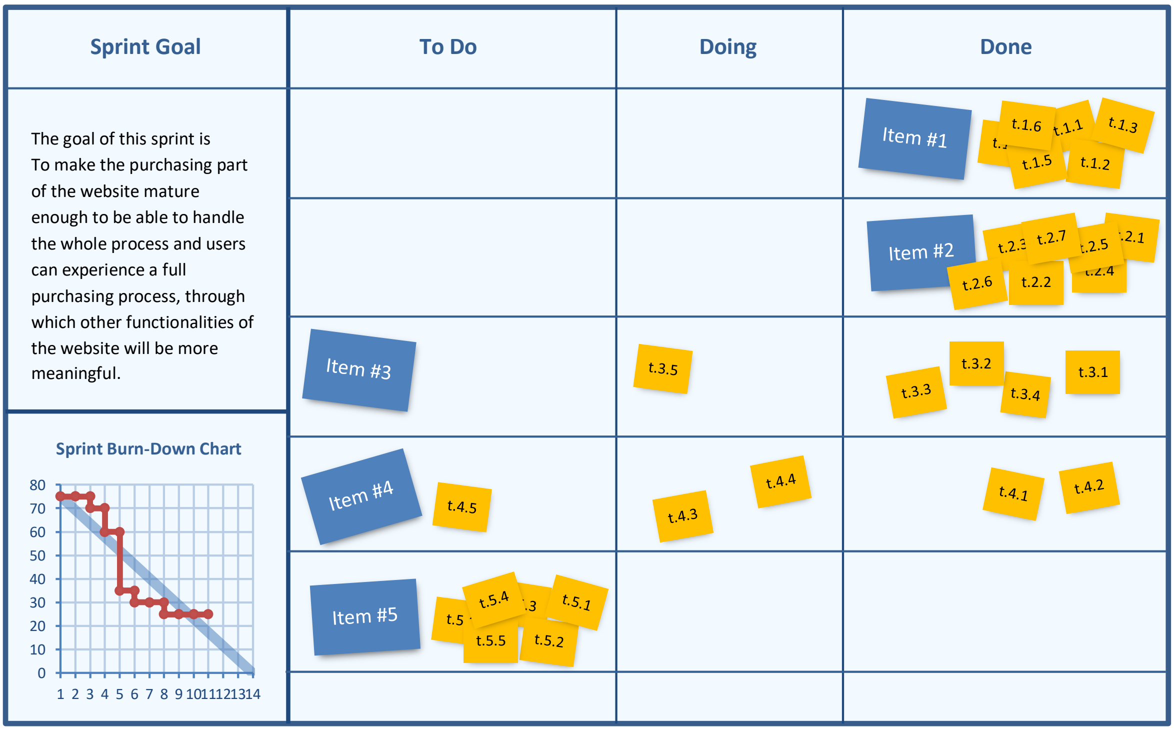
The vertical axis (remaining work) shows the amount of work (which is a sum of all the estimates for each item in the Product Backlog), and the horizontal axis shows the amount of time passed from the beginning of the project or the number of Sprints passed.

It's also common to add a trend-line to the burn-down chart, to get a rough estimate of the completion date. This estimate is based on the assumption that no new items will be added to the Product Backlog and the development capacity will stay the same. It's expected from a Product Owner to only use this as a guide, and come up with a more reliable forecast for the completion date.

4.6 Monitoring Sprint Progress

Besides the monitoring done for the whole project, we should also monitor the progress of each single Sprint throughout its life. This is the responsibility of the Development Team and should be done at least once per Daily Scrum.

This information is used to calculate the likelihood of achieving the Sprint Goal and completing all items of the Sprint Backlog.



The goal of this sprint is To make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.

The Sprint progress information can be represented by a burn-down chart, and this chart can be a part of the Sprint board, where everyone can see.

4.7 Velocity

Assume the following is the size and status of the Sprint Backlog items at the end of the Sprint:

- Item 1, size: 12, done
- Item 2, size: 8, done
- Item 3, size: 2, done
- Item 4, size: 6, 50% complete
- Item 5, size: 10, not done

The “size” here refers to the estimates done by the developers. In this case, what is the size of Sprint output?

It is 12+8+2=22 units.

Remember that we do not count the in-progress items. We do not even mention something like “50% complete”; it is just “not done”.

No, assume that the following is the outputs of the past Sprints:

- Sprint 1: 14 units
- Sprint 2: 19 units
- Sprint 3: 28 units
- Sprint 4: 23 units
- Sprint 5: 22 units

Now, fill in the following sentence:

On average, we complete … units per Sprint.

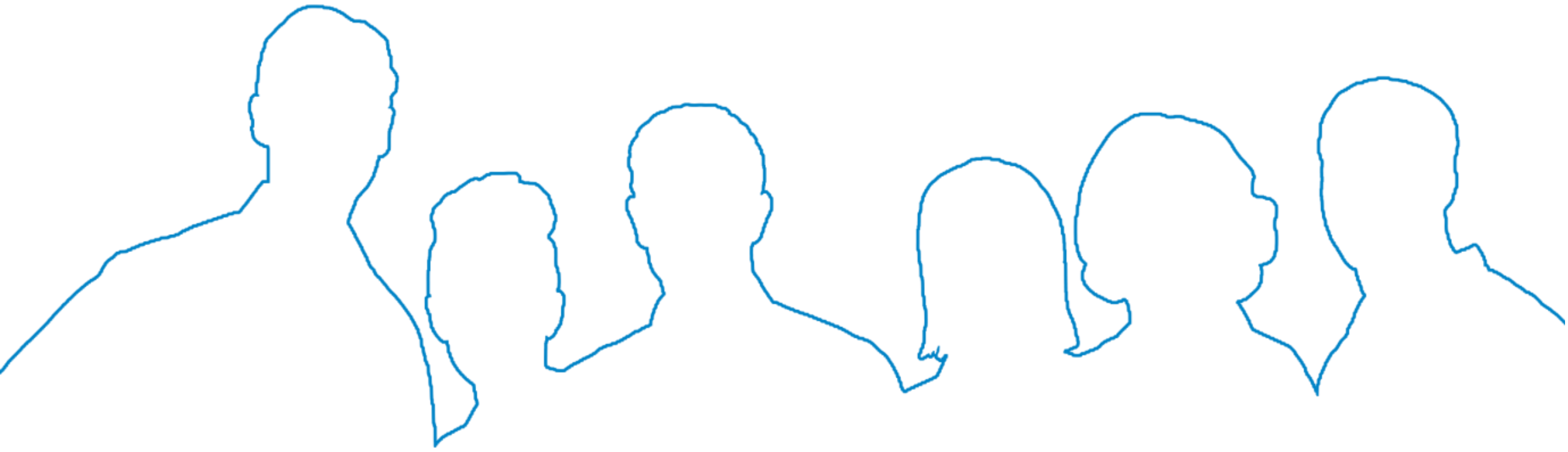
The answer is (14+19+28+23+22)/5≈21 units. This is what we call *velocity*:

At the moment, the velocity of our project is 21 units per Sprint.

Velocity is calculated by the Development Team, and usually helps for the following:

- Helping the developers decide how much work we can pick for the next Sprint
- Helping the Product Owner forecast how many Sprints it takes to develop the remaining items in the Product Backlog

What's Next?



5 What's Next?

We hope you have enjoyed this short, simple ebook. The goal was to explain the core Scrum framework, as defined in the Scrum Guide. Therefore, we did not go through techniques and practices used in Scrum projects (e.g. Story Points, Planning Poker, etc.) Before starting to use Scrum, you need to learn about those concepts as well.

In case you are interested in taking the PSM™ exam from Scrum.org, you can get help from our two products:

PSM™ I Simulated Exams

3 sets of exams with complete explanations for each question.

PSM™ I Preparation Online Course

A complete online course on the core Scrum framework that prepares you for the exam.

Samples of each product are available in the website.

If you are interested in learning Agile frameworks other than Scrum, you can check our free email courses.

Good luck ☺