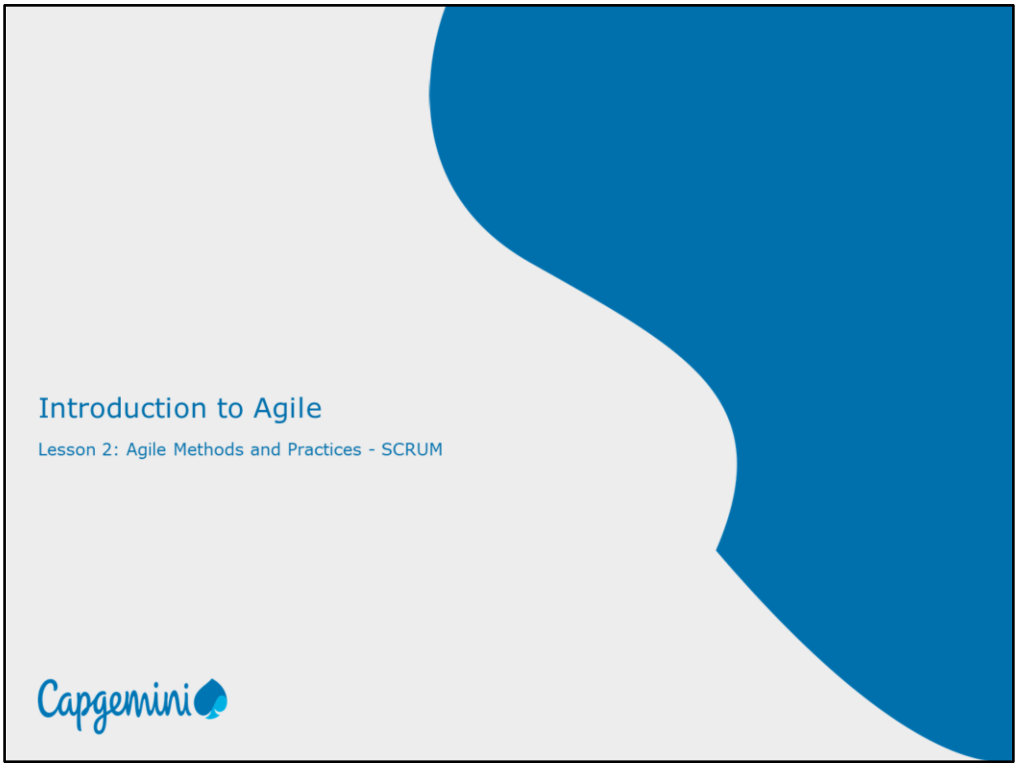
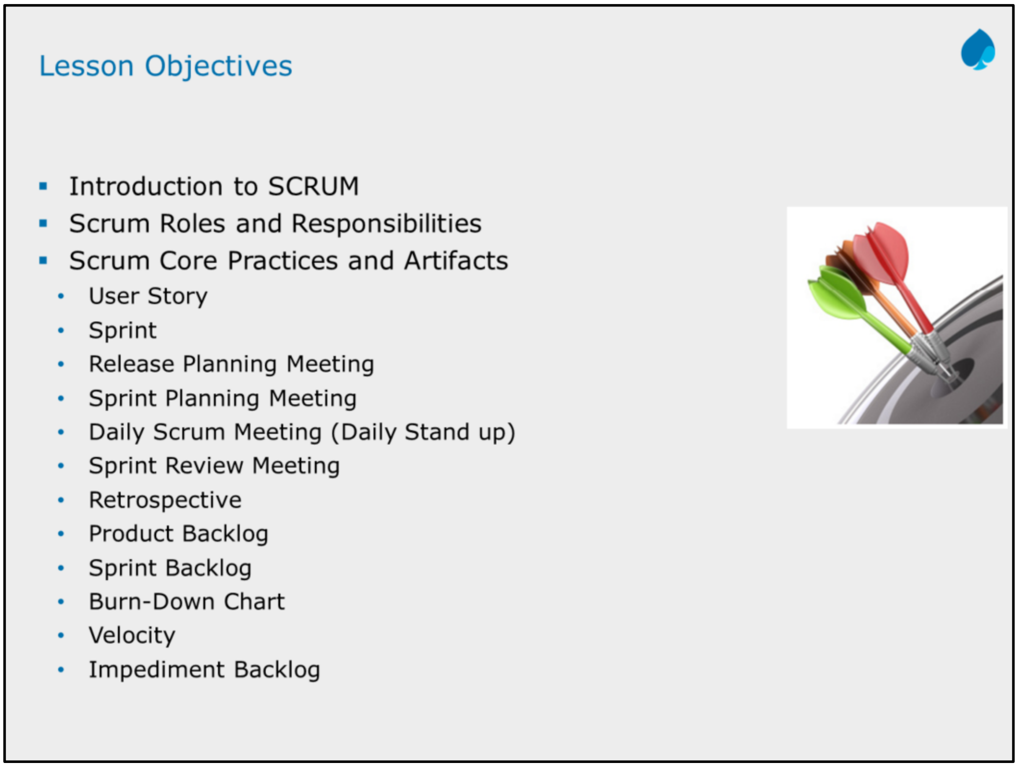


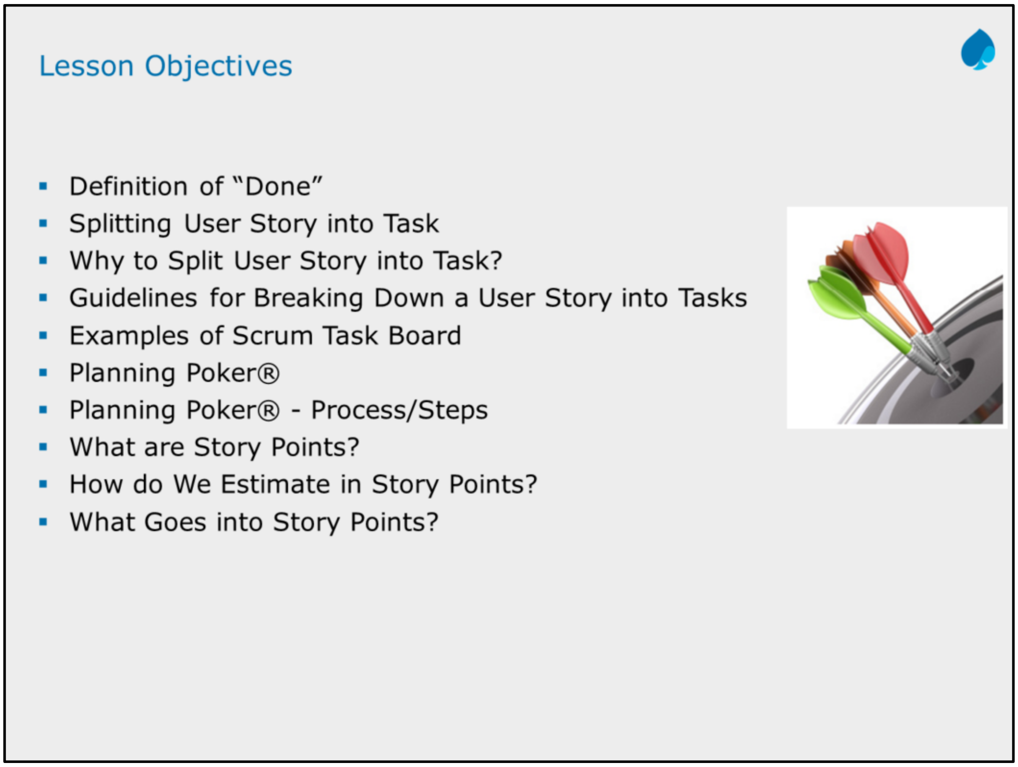
Introduction to Agile

Lesson 2: Agile Methods and Practices - SCRUM



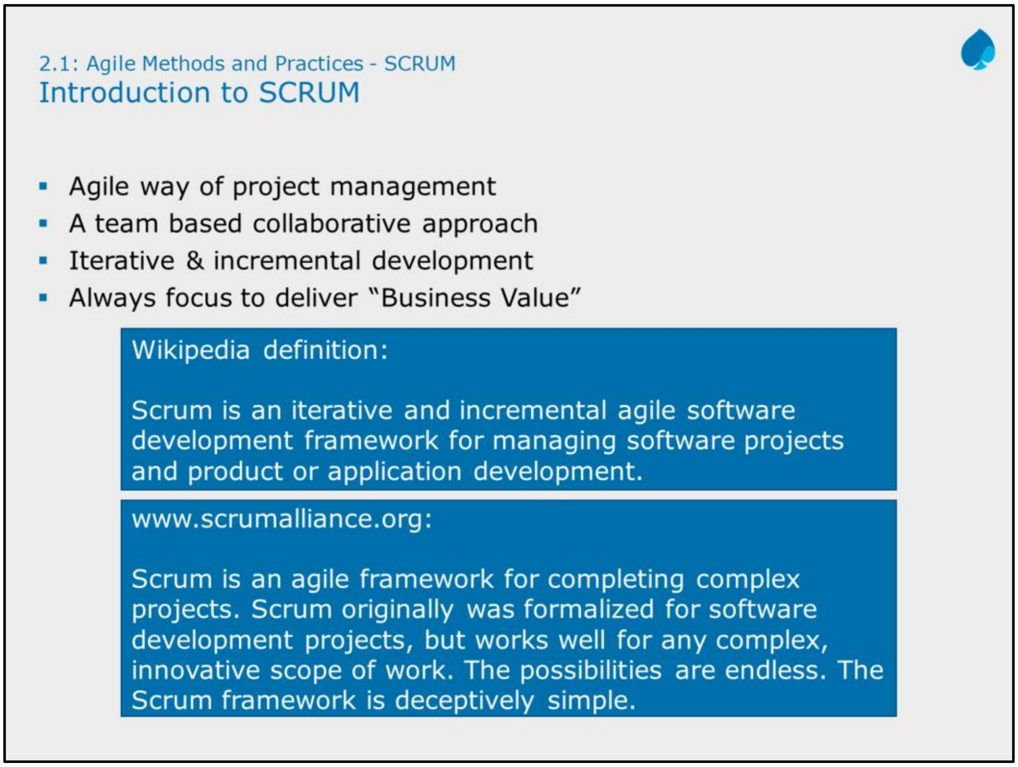
Lesson Objectives

- Introduction to SCRUM
- Scrum Roles and Responsibilities
- Scrum Core Practices and Artifacts
  - User Story
  - Sprint
  - Release Planning Meeting
  - Sprint Planning Meeting
  - Daily Scrum Meeting (Daily Stand up)
  - Sprint Review Meeting
  - Retrospective
  - Product Backlog
  - Sprint Backlog
  - Burn-Down Chart
  - Velocity
  - Impediment Backlog



Lesson Objectives

- Definition of "Done"
- Splitting User Story into Task
- Why to Split User Story into Task?
- Guidelines for Breaking Down a User Story into Tasks
- Examples of Scrum Task Board
- Planning Poker®
- Planning Poker® - Process/Steps
- What are Story Points?
- How do We Estimate in Story Points?
- What Goes into Story Points?



2.1: Agile Methods and Practices - SCRUM

Introduction to SCRUM

- Agile way of project management
- A team based collaborative approach
- Iterative & incremental development
- Always focus to deliver "Business Value"

Wikipedia definition:

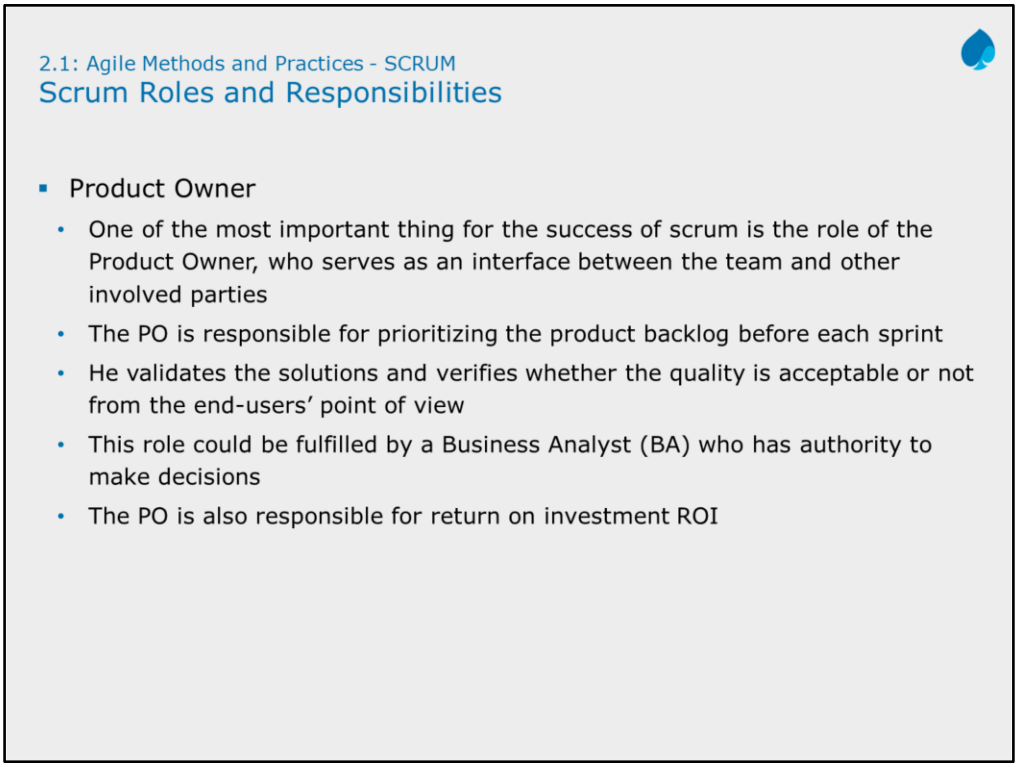
Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development.

www.scrumalliance.org:

Scrum is an agile framework for completing complex projects. Scrum originally was formalized for software development projects, but works well for any complex, innovative scope of work. The possibilities are endless. The Scrum framework is deceptively simple.

Introduction to SCRUM

Scrum is one of the more popular agile methods in use today and originally developed at Easel in 1993 by Jeff Sutherland and Ken Schwaber. Since that time it has been used at many software companies, which have resulted in the method being extended and enhanced.
When Jeff Sutherland created the scrum process in 1993, he borrowed the term "scrum" from an analogy put forth in a 1986 study by Takeuchi and Nonaka, published in the Harvard Business Review. In that study, Takeuchi and Nonaka compare high-performing, cross-functional teams to the scrum formation used by Rugby teams.
Scrum is the leading agile development methodology, used by Fortune 500 companies around the world. The Scrum Alliance exists to transform the way we tackle complex projects, bringing the Scrum framework and agile principles beyond software development to the broader world of work.



2.1: Agile Methods and Practices - SCRUM

Scrum Roles and Responsibilities

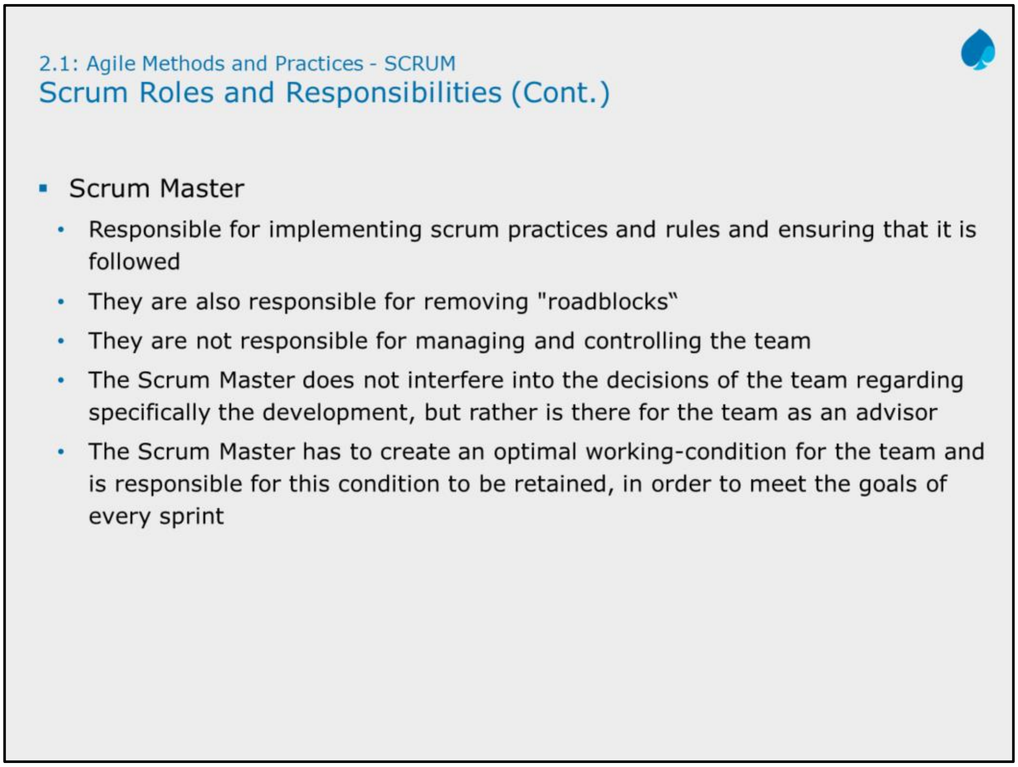
- Product Owner
  - One of the most important thing for the success of scrum is the role of the Product Owner, who serves as an interface between the team and other involved parties
  - The PO is responsible for prioritizing the product backlog before each sprint
  - He validates the solutions and verifies whether the quality is acceptable or not from the end-users' point of view
  - This role could be fulfilled by a Business Analyst (BA) who has authority to make decisions
  - The PO is also responsible for return on investment ROI

Scrum Roles and Responsibilities:

In contrast to classical project management methods, Scrum doesn't have and doesn't need a product manager, a task manager or a team leader. The most important three roles of Scrum are:

Product Owner
Scrum Master
Development team

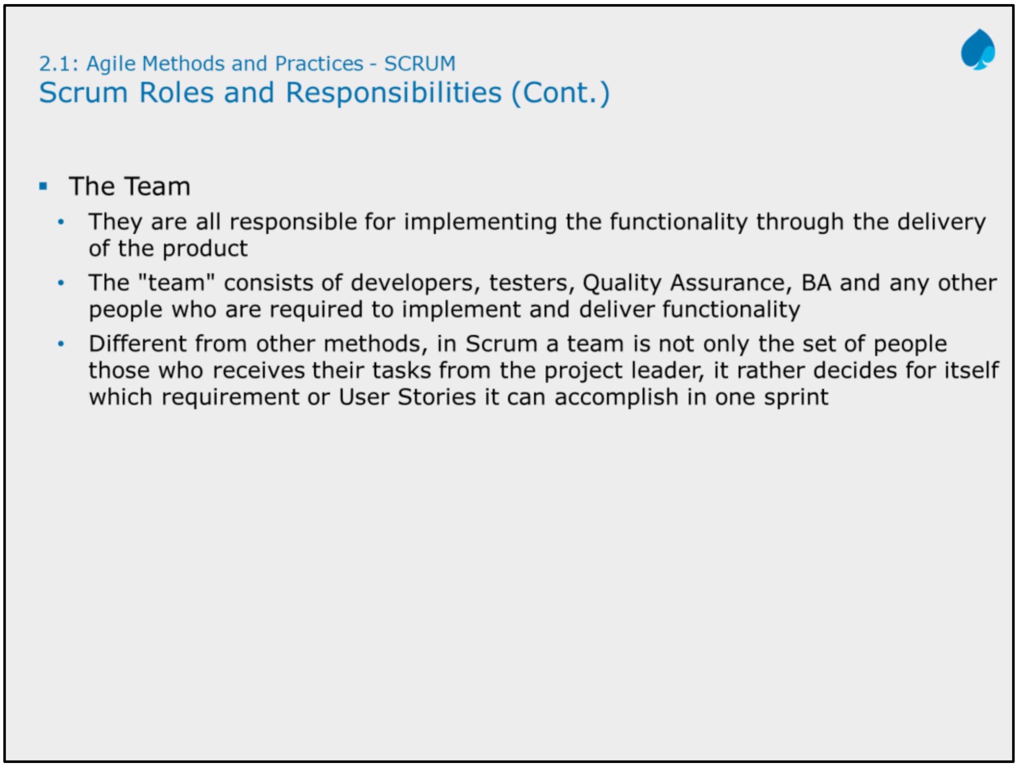
These three roles are coequal and all of them have certain responsibilities.



2.1: Agile Methods and Practices - SCRUM

Scrum Roles and Responsibilities (Cont.)

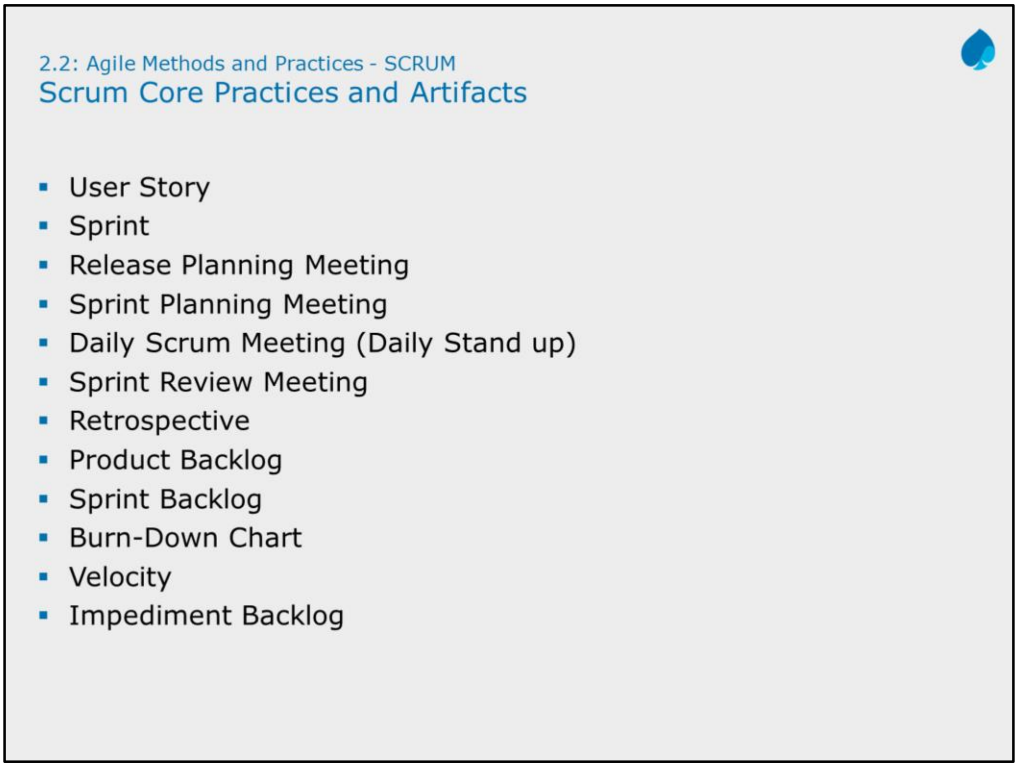
- Scrum Master
  - Responsible for implementing scrum practices and rules and ensuring that it is followed
  - They are also responsible for removing "roadblocks‟
  - They are not responsible for managing and controlling the team
  - The Scrum Master does not interfere into the decisions of the team regarding specifically the development, but rather is there for the team as an advisor
  - The Scrum Master has to create an optimal working-condition for the team and is responsible for this condition to be retained, in order to meet the goals of every sprint



2.1: Agile Methods and Practices - SCRUM

Scrum Roles and Responsibilities (Cont.)

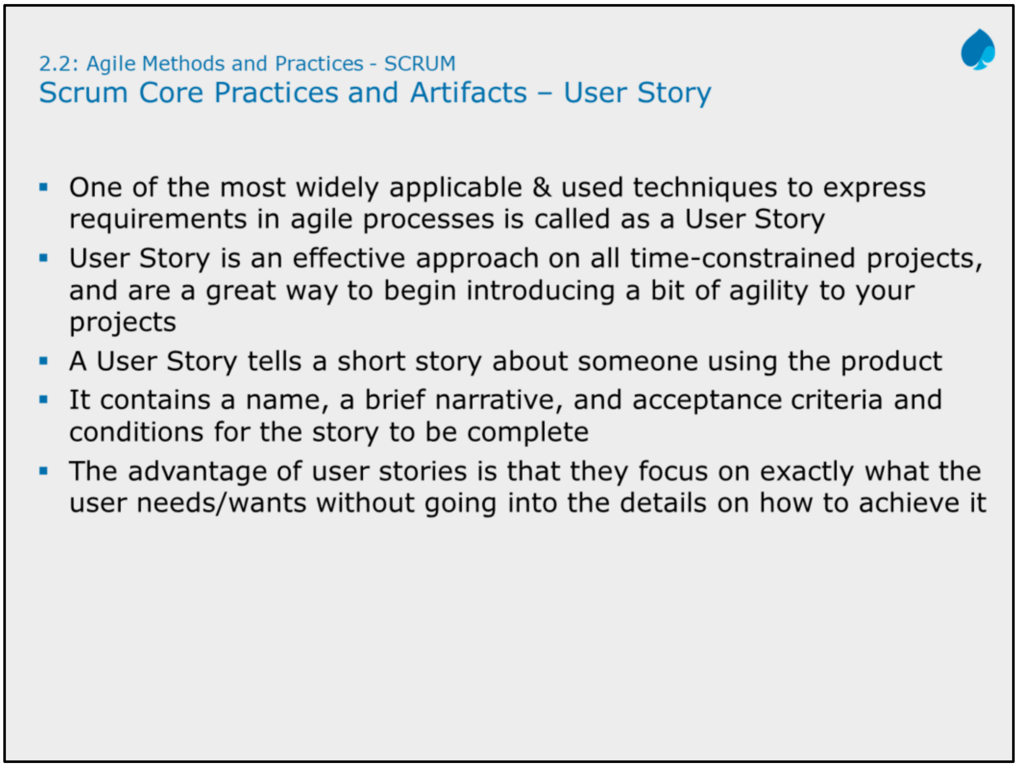
- The Team
  - They are all responsible for implementing the functionality through the delivery of the product
  - The "team" consists of developers, testers, Quality Assurance, BA and any other people who are required to implement and deliver functionality
  - Different from other methods, in Scrum a team is not only the set of people those who receives their tasks from the project leader, it rather decides for itself which requirement or User Stories it can accomplish in one sprint



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts

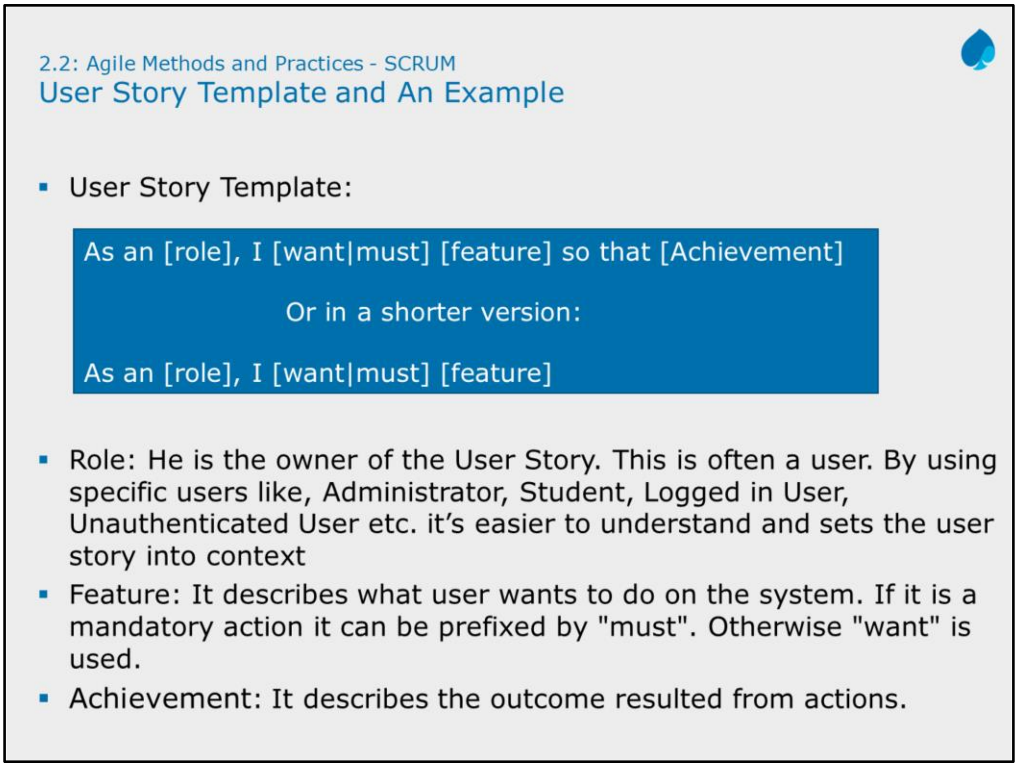
- User Story
- Sprint
- Release Planning Meeting
- Sprint Planning Meeting
- Daily Scrum Meeting (Daily Stand up)
- Sprint Review Meeting
- Retrospective
- Product Backlog
- Sprint Backlog
- Burn-Down Chart
- Velocity
- Impediment Backlog



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – User Story

- One of the most widely applicable & used techniques to express requirements in agile processes is called as a User Story
- User Story is an effective approach on all time-constrained projects, and are a great way to begin introducing a bit of agility to your projects
- A User Story tells a short story about someone using the product
- It contains a name, a brief narrative, and acceptance criteria and conditions for the story to be complete
- The advantage of user stories is that they focus on exactly what the user needs/wants without going into the details on how to achieve it



2.2: Agile Methods and Practices - SCRUM

User Story Template and An Example

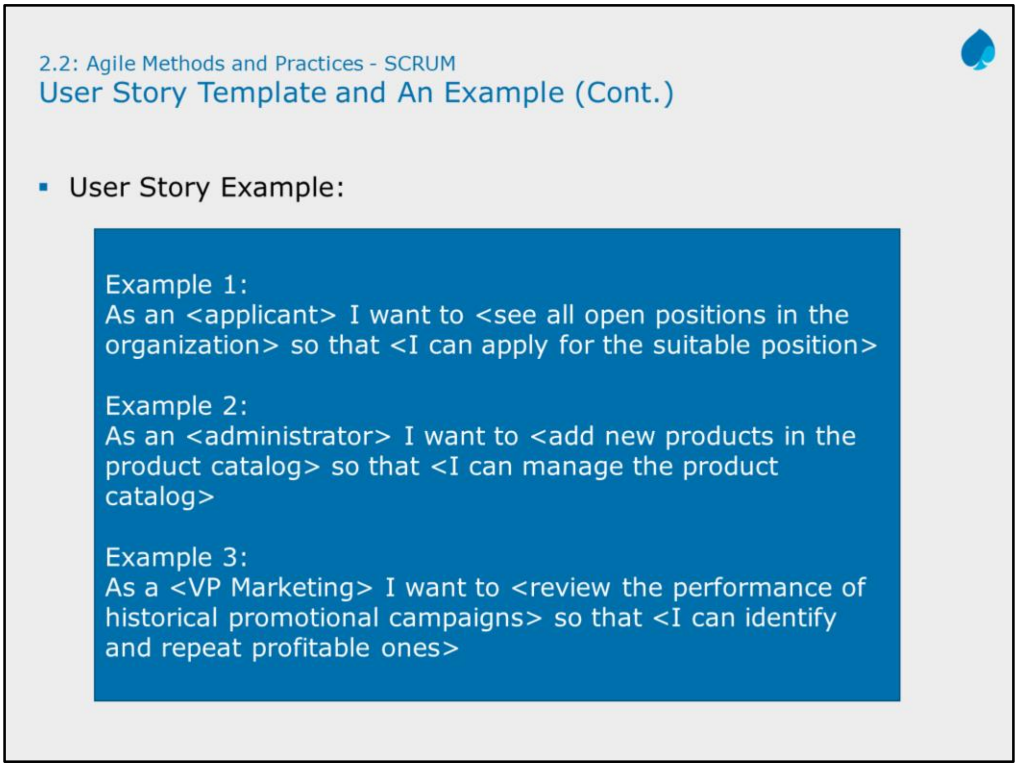
- User Story Template:

  As an [role], I [want|must] [feature] so that [Achievement]

  Or in a shorter version:

  As an [role], I [want|must] [feature]

- Role: He is the owner of the User Story. This is often a user. By using specific users like, Administrator, Student, Logged in User, Unauthenticated User etc. it's easier to understand and sets the user story into context
- Feature: It describes what user wants to do on the system. If it is a mandatory action it can be prefixed by "must". Otherwise "want" is used.
- Achievement: It describes the outcome resulted from actions.



2.2: Agile Methods and Practices - SCRUM

User Story Template and An Example (Cont.)

- User Story Example:

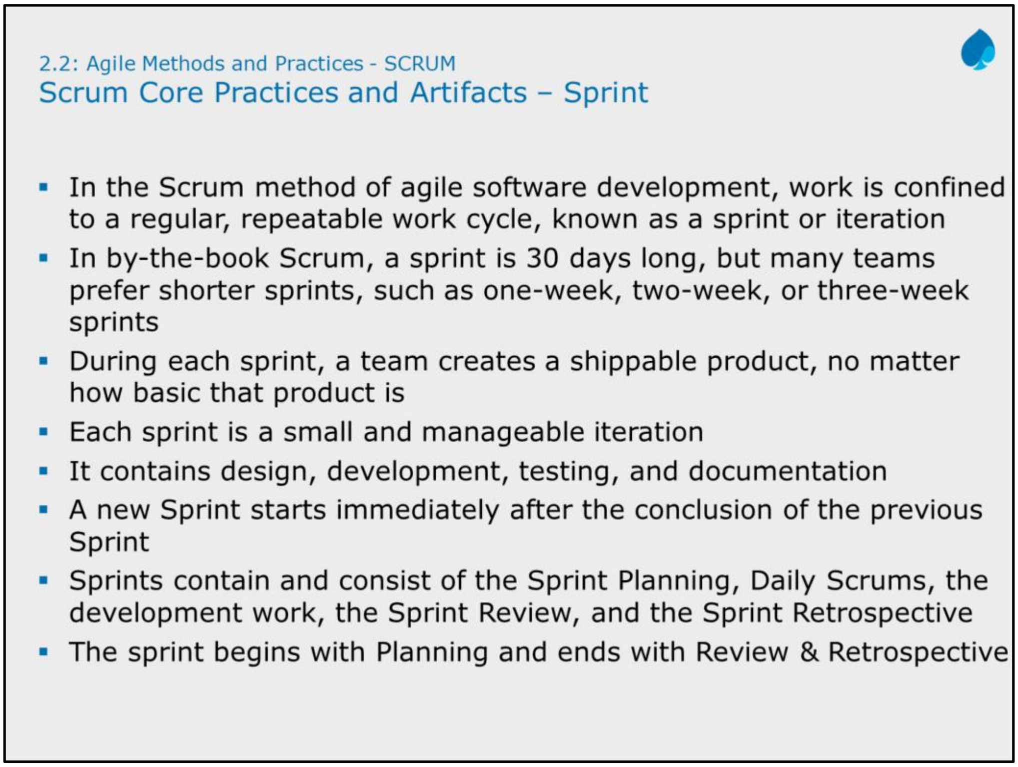
Example 1:
As an <applicant> I want to <see all open positions in the organization> so that <I can apply for the suitable position>

Example 2:
As an <administrator> I want to <add new products in the product catalog> so that <I can manage the product catalog>

Example 3:
As a <VP Marketing> I want to <review the performance of historical promotional campaigns> so that <I can identify and repeat profitable ones>

Class Exercise:

- Writing User Stories



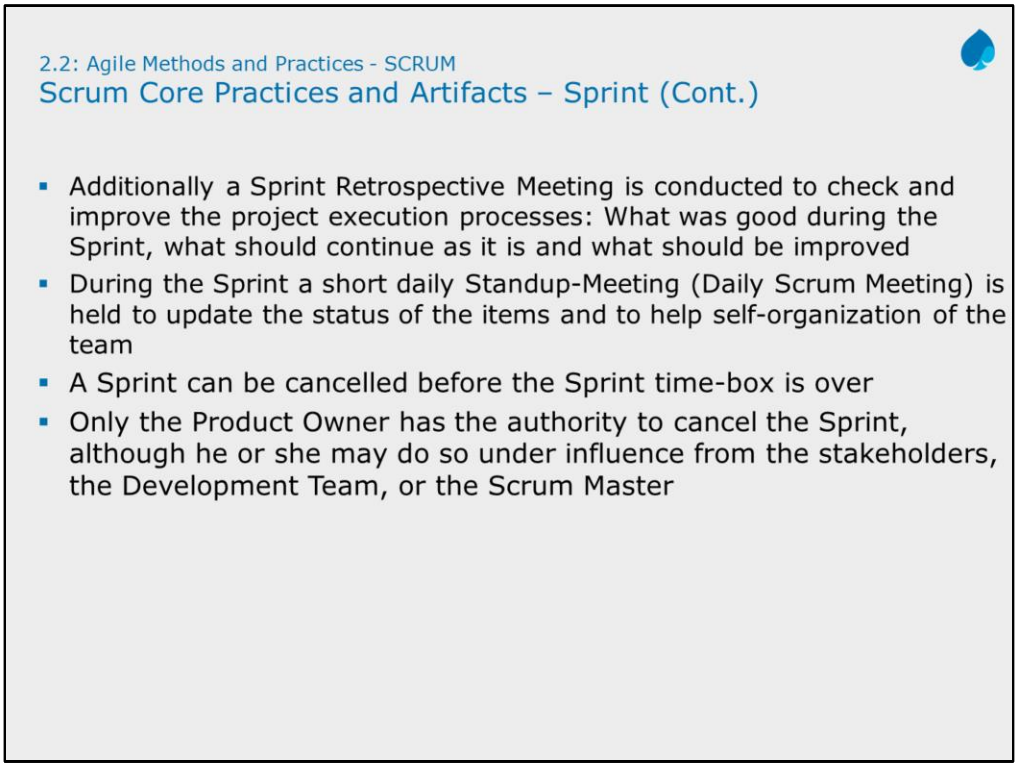
2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Sprint

- In the Scrum method of agile software development, work is confined to a regular, repeatable work cycle, known as a sprint or iteration
- In by-the-book Scrum, a sprint is 30 days long, but many teams prefer shorter sprints, such as one-week, two-week, or three-week sprints
- During each sprint, a team creates a shippable product, no matter how basic that product is
- Each sprint is a small and manageable iteration
- It contains design, development, testing, and documentation
- A new Sprint starts immediately after the conclusion of the previous Sprint
- Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective
- The sprint begins with Planning and ends with Review & Retrospective

Introduction to Sprint

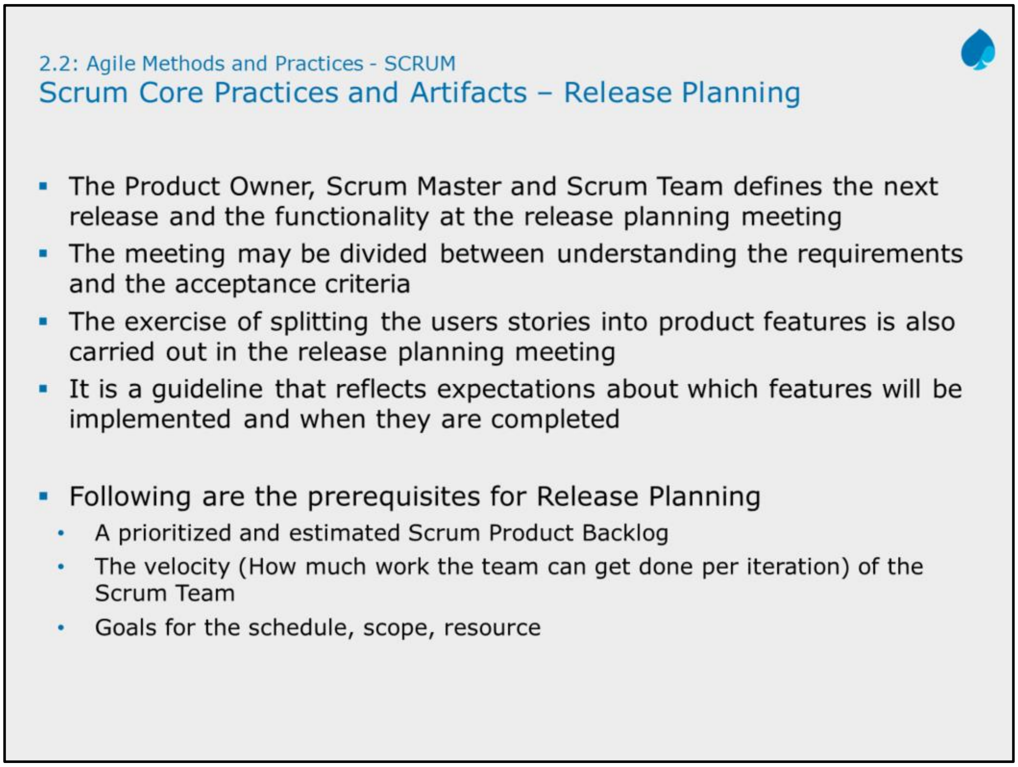
Each Sprint may be considered a project with no more than a one-month duration. Like projects, Sprints are used to accomplish something. Each Sprint has a goal of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product increment.
A Sprint would be cancelled if the Sprint Goal becomes obsolete. This might occur if the company changes direction or if market or technology conditions change. In general, a Sprint should be cancelled if it no longer makes sense given the circumstances. But, due to the short duration of Sprints, cancellation rarely makes sense.



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Sprint (Cont.)

- Additionally a Sprint Retrospective Meeting is conducted to check and improve the project execution processes: What was good during the Sprint, what should continue as it is and what should be improved
- During the Sprint a short daily Standup-Meeting (Daily Scrum Meeting) is held to update the status of the items and to help self-organization of the team
- A Sprint can be cancelled before the Sprint time-box is over
- Only the Product Owner has the authority to cancel the Sprint, although he or she may do so under influence from the stakeholders, the Development Team, or the Scrum Master

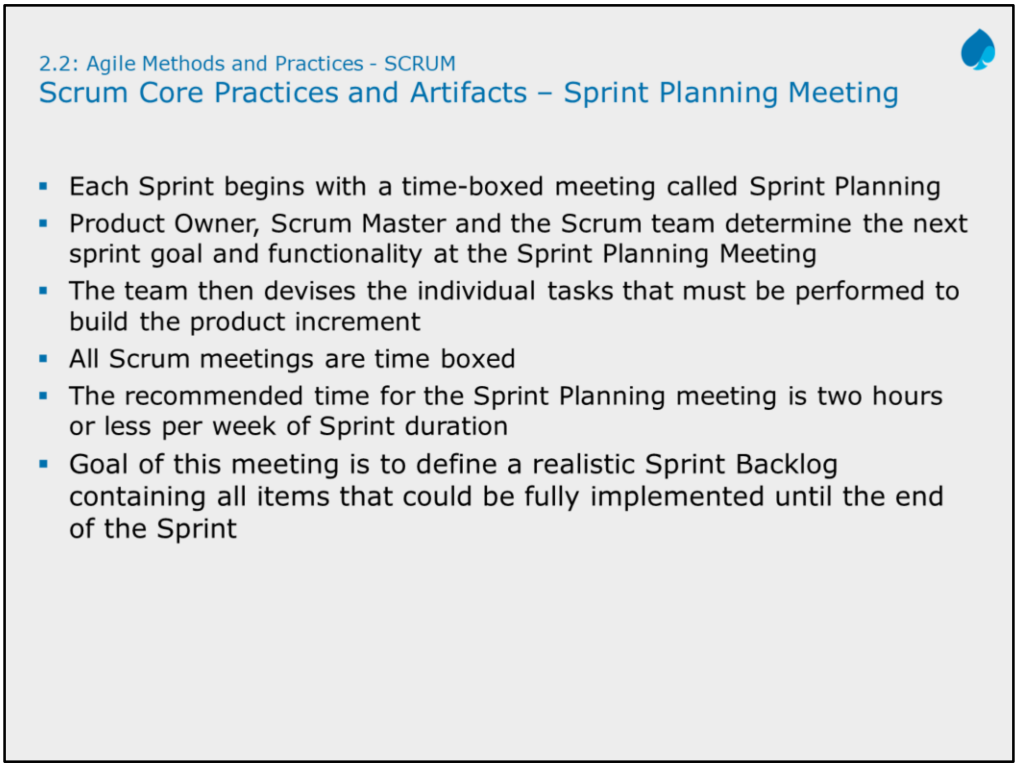


2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Release Planning

- The Product Owner, Scrum Master and Scrum Team defines the next release and the functionality at the release planning meeting
- The meeting may be divided between understanding the requirements and the acceptance criteria
- The exercise of splitting the users stories into product features is also carried out in the release planning meeting
- It is a guideline that reflects expectations about which features will be implemented and when they are completed

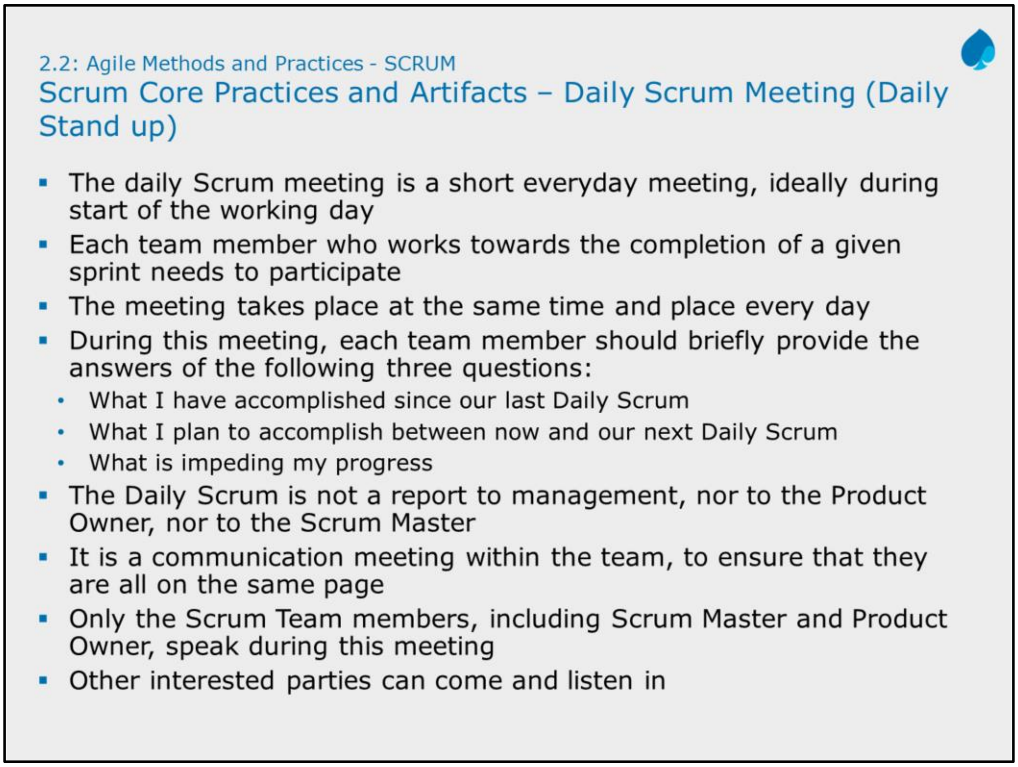
- Following are the prerequisites for Release Planning
  - A prioritized and estimated Scrum Product Backlog
  - The velocity (How much work the team can get done per iteration) of the Scrum Team
  - Goals for the schedule, scope, resource



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Sprint Planning Meeting

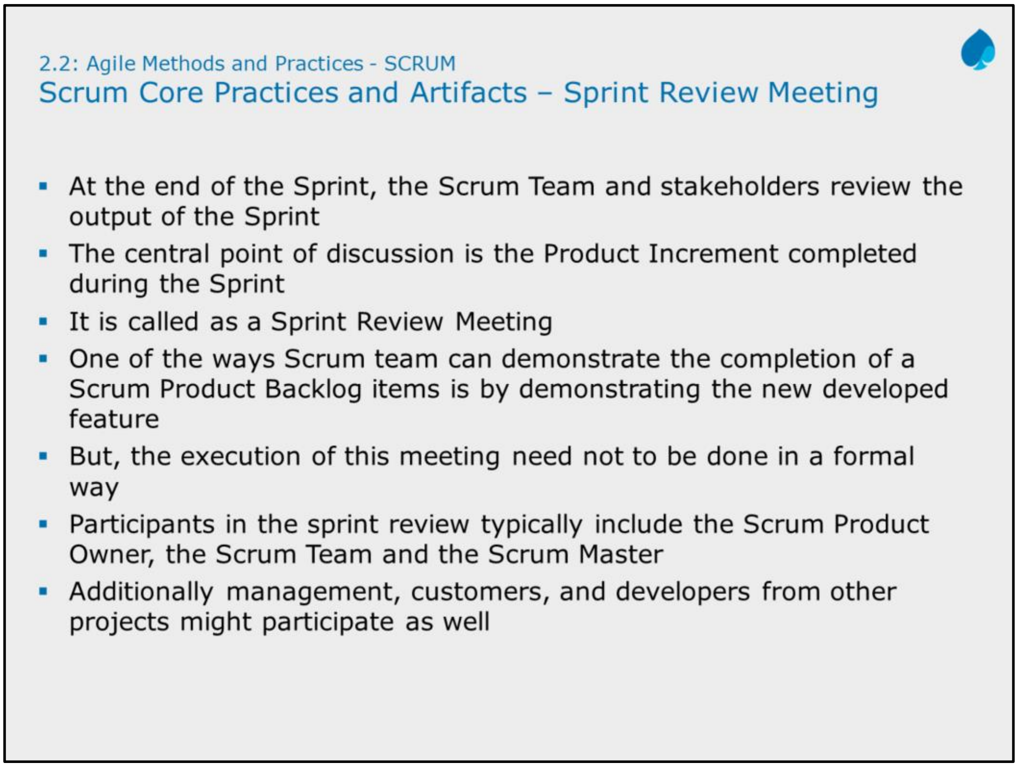
- Each Sprint begins with a time-boxed meeting called Sprint Planning
- Product Owner, Scrum Master and the Scrum team determine the next sprint goal and functionality at the Sprint Planning Meeting
- The team then devises the individual tasks that must be performed to build the product increment
- All Scrum meetings are time boxed
- The recommended time for the Sprint Planning meeting is two hours or less per week of Sprint duration
- Goal of this meeting is to define a realistic Sprint Backlog containing all items that could be fully implemented until the end of the Sprint



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Daily Scrum Meeting (Daily Stand up)

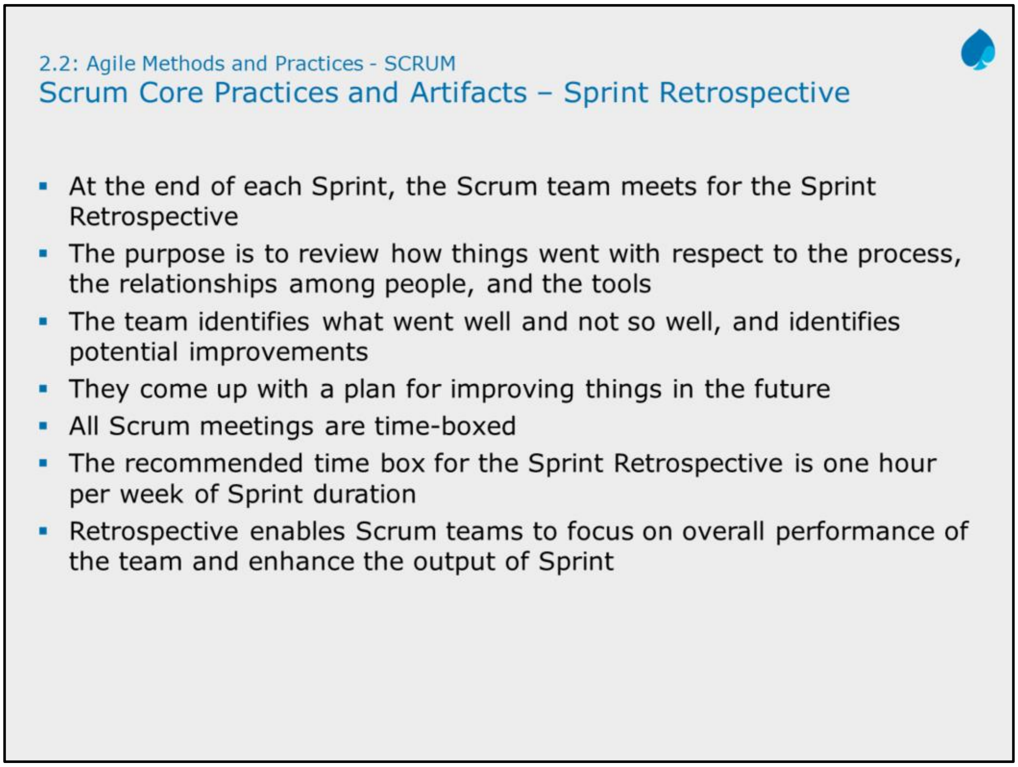
- The daily Scrum meeting is a short everyday meeting, ideally during start of the working day
- Each team member who works towards the completion of a given sprint needs to participate
- The meeting takes place at the same time and place every day
- During this meeting, each team member should briefly provide the answers of the following three questions:
  - What I have accomplished since our last Daily Scrum
  - What I plan to accomplish between now and our next Daily Scrum
  - What is impeding my progress
- The Daily Scrum is not a report to management, nor to the Product Owner, nor to the Scrum Master
- It is a communication meeting within the team, to ensure that they are all on the same page
- Only the Scrum Team members, including Scrum Master and Product Owner, speak during this meeting
- Other interested parties can come and listen in



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Sprint Review Meeting

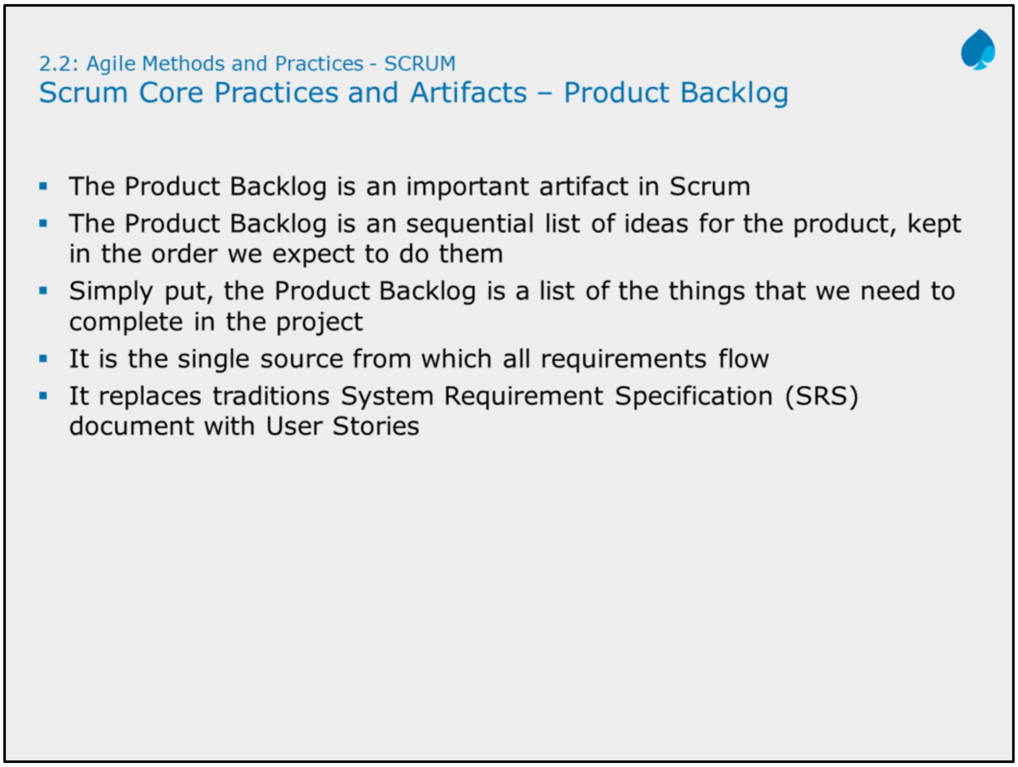
- At the end of the Sprint, the Scrum Team and stakeholders review the output of the Sprint
- The central point of discussion is the Product Increment completed during the Sprint
- It is called as a Sprint Review Meeting
- One of the ways Scrum team can demonstrate the completion of a Scrum Product Backlog items is by demonstrating the new developed feature
- But, the execution of this meeting need not to be done in a formal way
- Participants in the sprint review typically include the Scrum Product Owner, the Scrum Team and the Scrum Master
- Additionally management, customers, and developers from other projects might participate as well



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Sprint Retrospective

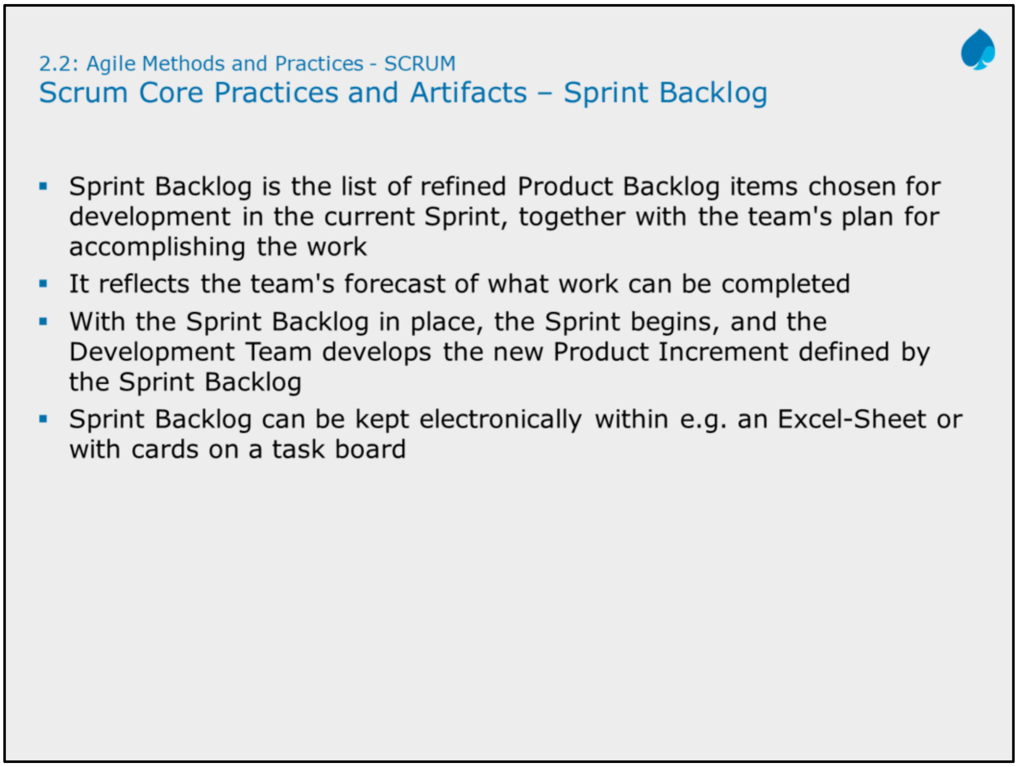
- At the end of each Sprint, the Scrum team meets for the Sprint Retrospective
- The purpose is to review how things went with respect to the process, the relationships among people, and the tools
- The team identifies what went well and not so well, and identifies potential improvements
- They come up with a plan for improving things in the future
- All Scrum meetings are time-boxed
- The recommended time box for the Sprint Retrospective is one hour per week of Sprint duration
- Retrospective enables Scrum teams to focus on overall performance of the team and enhance the output of Sprint



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Product Backlog

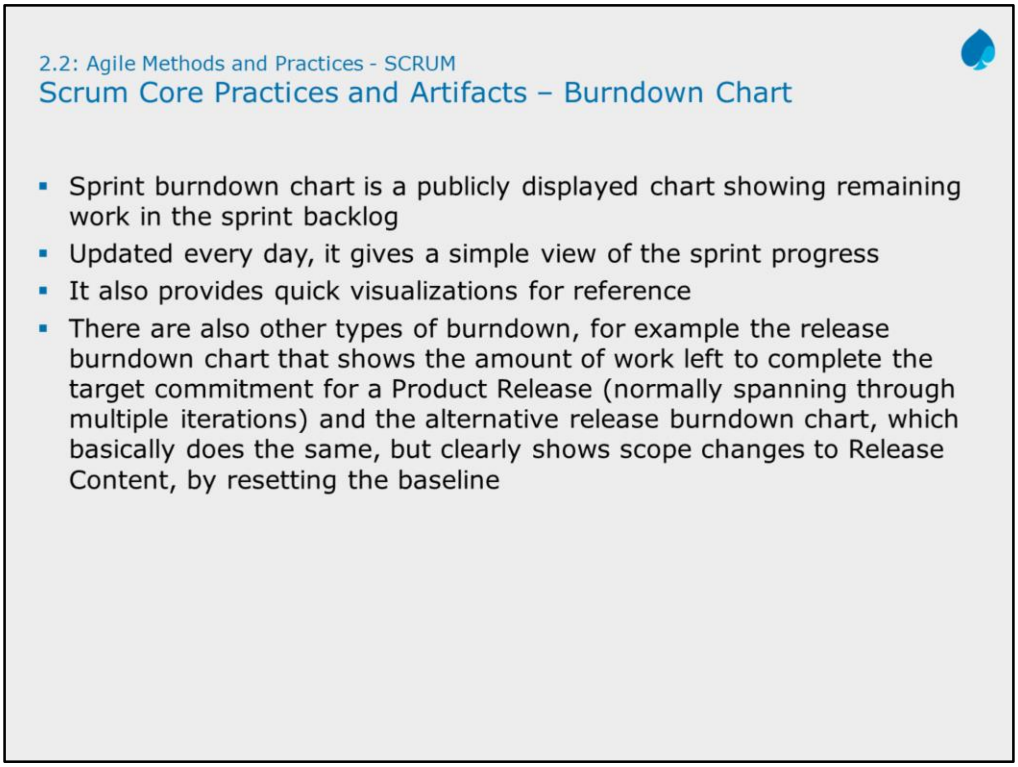
- The Product Backlog is an important artifact in Scrum
- The Product Backlog is an sequential list of ideas for the product, kept in the order we expect to do them
- Simply put, the Product Backlog is a list of the things that we need to complete in the project
- It is the single source from which all requirements flow
- It replaces traditions System Requirement Specification (SRS) document with User Stories



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Sprint Backlog

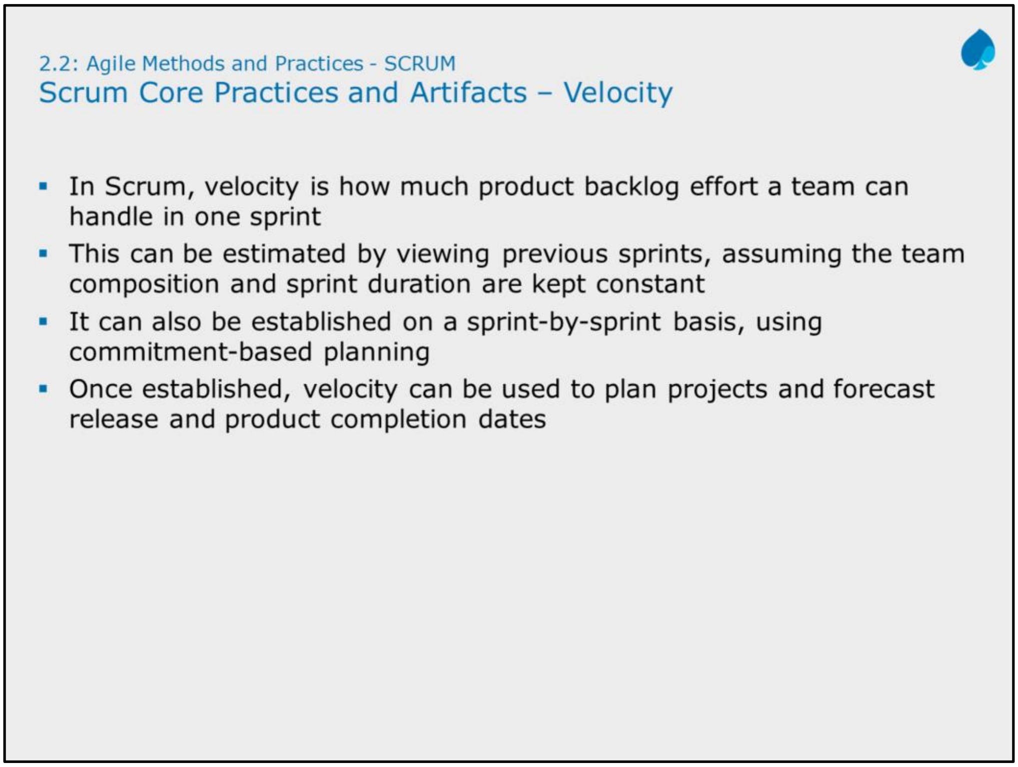
- Sprint Backlog is the list of refined Product Backlog items chosen for development in the current Sprint, together with the team's plan for accomplishing the work
- It reflects the team's forecast of what work can be completed
- With the Sprint Backlog in place, the Sprint begins, and the Development Team develops the new Product Increment defined by the Sprint Backlog
- Sprint Backlog can be kept electronically within e.g. an Excel-Sheet or with cards on a task board



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Burndown Chart

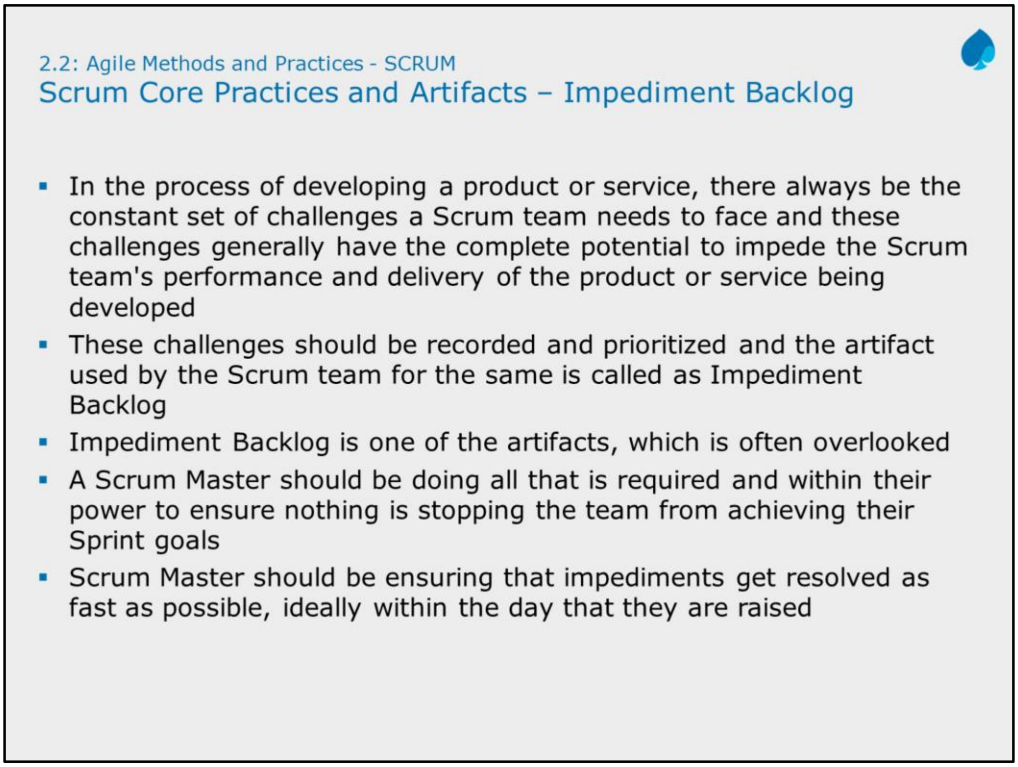
- Sprint burndown chart is a publicly displayed chart showing remaining work in the sprint backlog
- Updated every day, it gives a simple view of the sprint progress
- It also provides quick visualizations for reference
- There are also other types of burndown, for example the release burndown chart that shows the amount of work left to complete the target commitment for a Product Release (normally spanning through multiple iterations) and the alternative release burndown chart, which basically does the same, but clearly shows scope changes to Release Content, by resetting the baseline



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Velocity

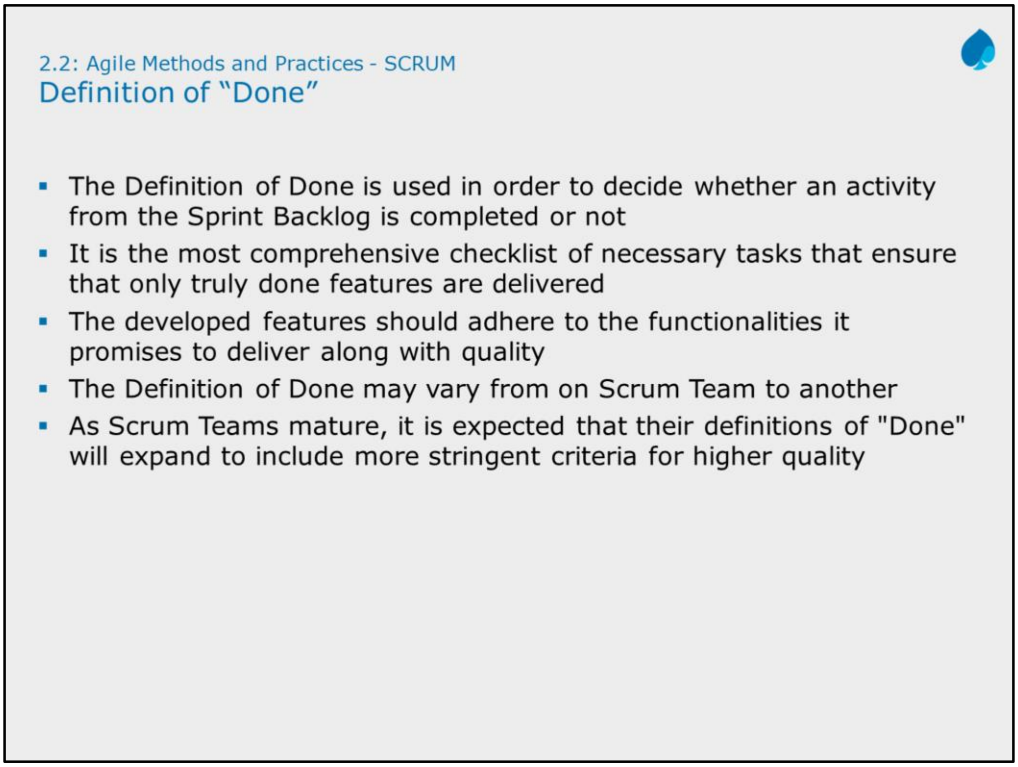
- In Scrum, velocity is how much product backlog effort a team can handle in one sprint
- This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant
- It can also be established on a sprint-by-sprint basis, using commitment-based planning
- Once established, velocity can be used to plan projects and forecast release and product completion dates



2.2: Agile Methods and Practices - SCRUM

Scrum Core Practices and Artifacts – Impediment Backlog

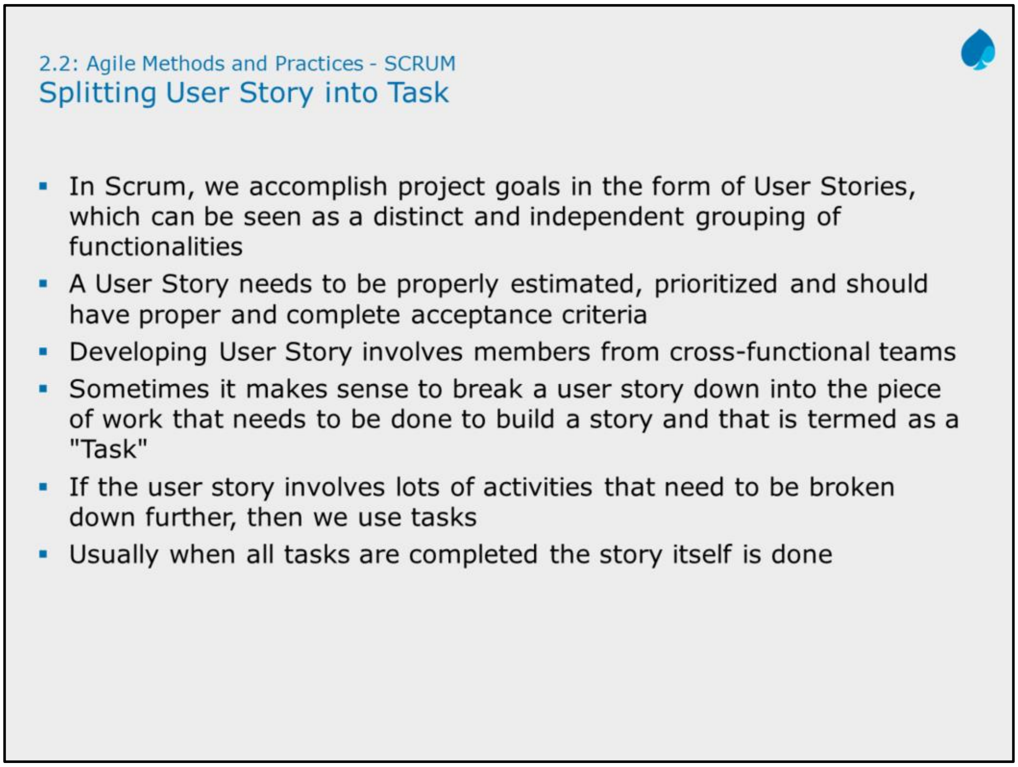
- In the process of developing a product or service, there always be the constant set of challenges a Scrum team needs to face and these challenges generally have the complete potential to impede the Scrum team's performance and delivery of the product or service being developed
- These challenges should be recorded and prioritized and the artifact used by the Scrum team for the same is called as Impediment Backlog
- Impediment Backlog is one of the artifacts, which is often overlooked
- A Scrum Master should be doing all that is required and within their power to ensure nothing is stopping the team from achieving their Sprint goals
- Scrum Master should be ensuring that impediments get resolved as fast as possible, ideally within the day that they are raised



2.2: Agile Methods and Practices - SCRUM

Definition of “Done”

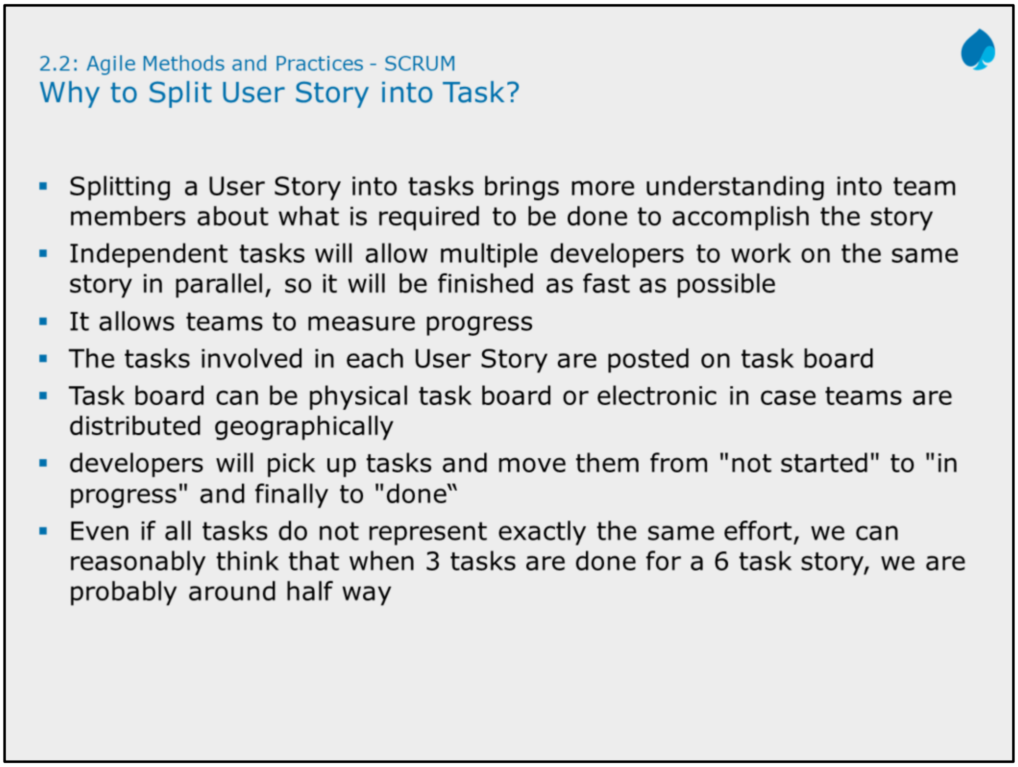
- The Definition of Done is used in order to decide whether an activity from the Sprint Backlog is completed or not
- It is the most comprehensive checklist of necessary tasks that ensure that only truly done features are delivered
- The developed features should adhere to the functionalities it promises to deliver along with quality
- The Definition of Done may vary from on Scrum Team to another
- As Scrum Teams mature, it is expected that their definitions of "Done" will expand to include more stringent criteria for higher quality



2.2: Agile Methods and Practices - SCRUM

Splitting User Story into Task

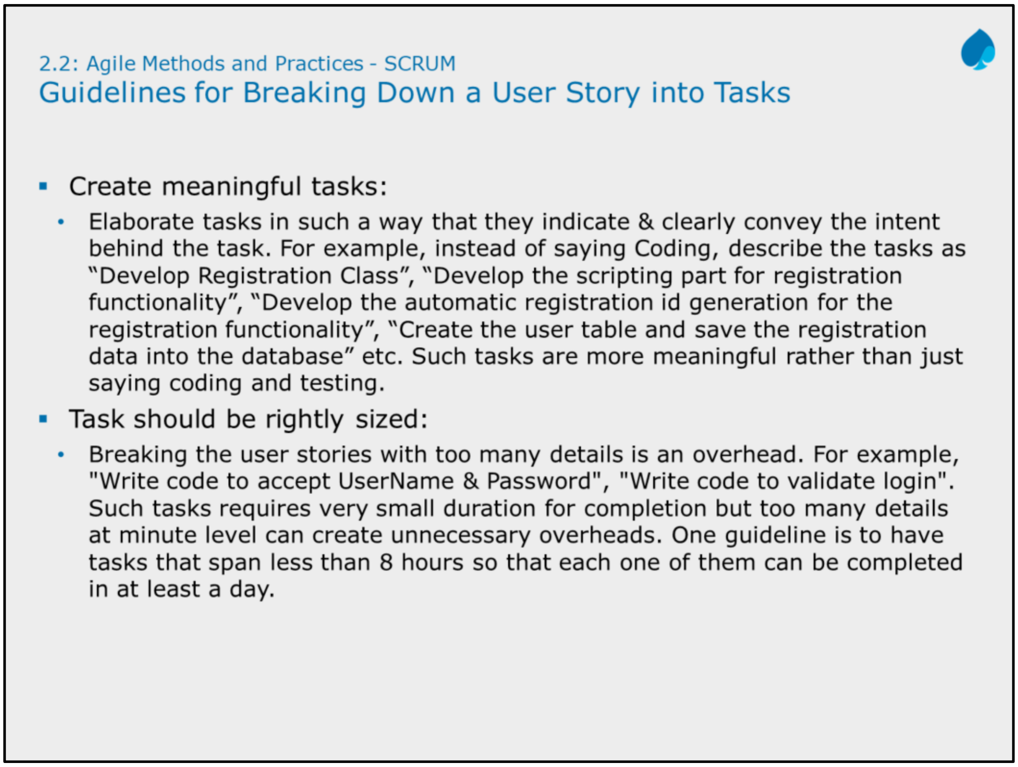
- In Scrum, we accomplish project goals in the form of User Stories, which can be seen as a distinct and independent grouping of functionalities
- A User Story needs to be properly estimated, prioritized and should have proper and complete acceptance criteria
- Developing User Story involves members from cross-functional teams
- Sometimes it makes sense to break a user story down into the piece of work that needs to be done to build a story and that is termed as a "Task"
- If the user story involves lots of activities that need to be broken down further, then we use tasks
- Usually when all tasks are completed the story itself is done



2.2: Agile Methods and Practices - SCRUM

Why to Split User Story into Task?

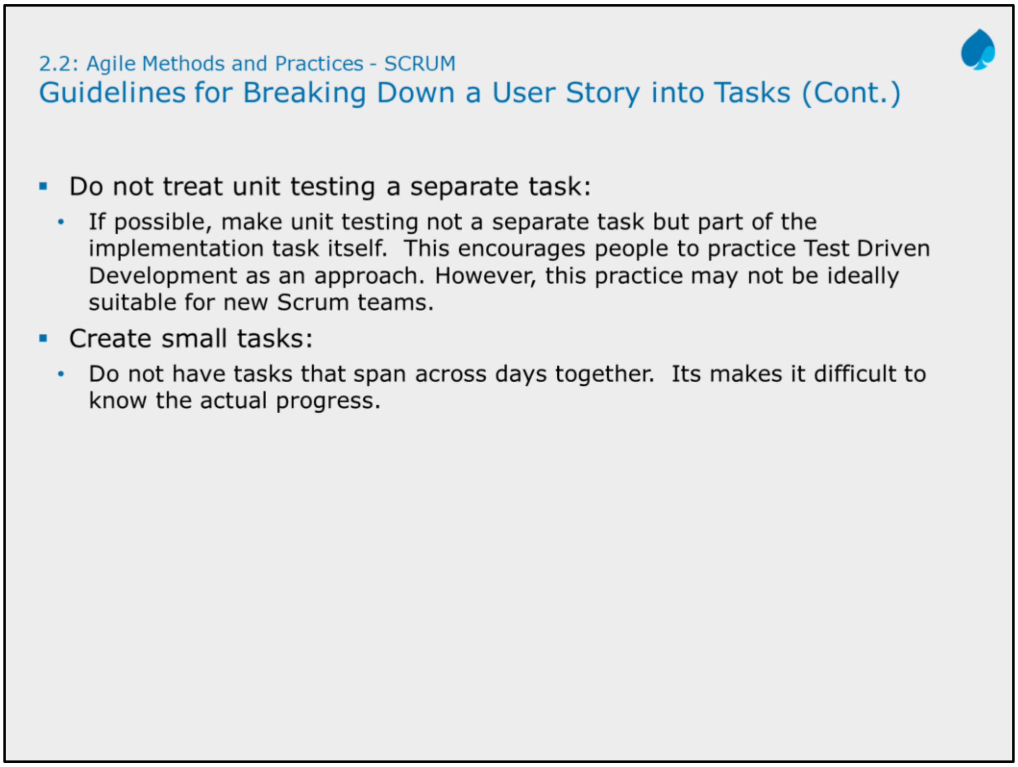
- Splitting a User Story into tasks brings more understanding into team members about what is required to be done to accomplish the story
- Independent tasks will allow multiple developers to work on the same story in parallel, so it will be finished as fast as possible
- It allows teams to measure progress
- The tasks involved in each User Story are posted on task board
- Task board can be physical task board or electronic in case teams are distributed geographically
- developers will pick up tasks and move them from "not started" to "in progress" and finally to "done‟
- Even if all tasks do not represent exactly the same effort, we can reasonably think that when 3 tasks are done for a 6 task story, we are probably around half way



2.2: Agile Methods and Practices - SCRUM

Guidelines for Breaking Down a User Story into Tasks

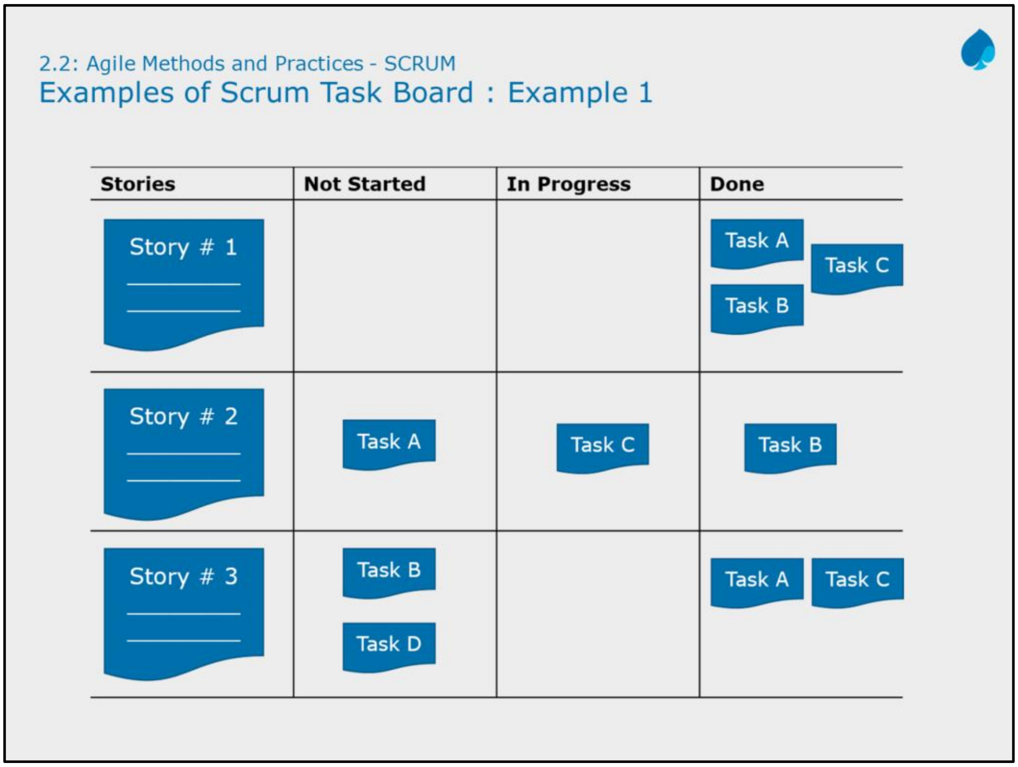
- Create meaningful tasks:
  - Elaborate tasks in such a way that they indicate & clearly convey the intent behind the task. For example, instead of saying Coding, describe the tasks as “Develop Registration Class”, “Develop the scripting part for registration functionality”, “Develop the automatic registration id generation for the registration functionality”, “Create the user table and save the registration data into the database” etc. Such tasks are more meaningful rather than just saying coding and testing.
- Task should be rightly sized:
  - Breaking the user stories with too many details is an overhead. For example, "Write code to accept UserName & Password", "Write code to validate login". Such tasks requires very small duration for completion but too many details at minute level can create unnecessary overheads. One guideline is to have tasks that span less than 8 hours so that each one of them can be completed in at least a day.



2.2: Agile Methods and Practices - SCRUM

Guidelines for Breaking Down a User Story into Tasks (Cont.)

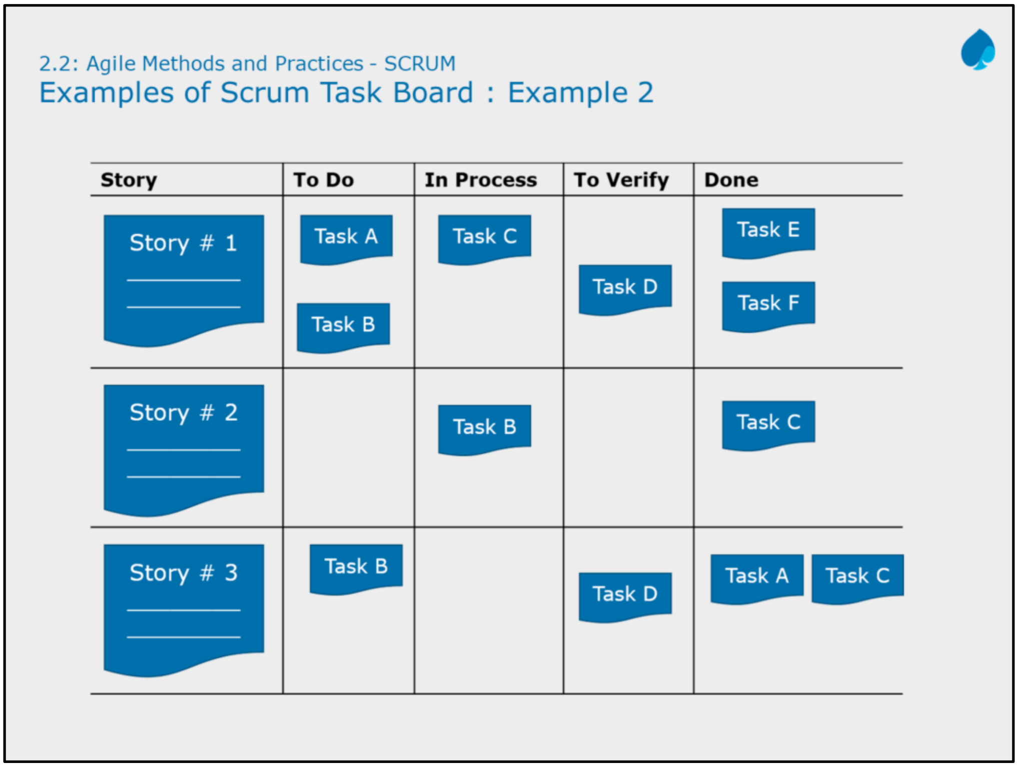
- Do not treat unit testing a separate task:
  - If possible, make unit testing not a separate task but part of the implementation task itself. This encourages people to practice Test Driven Development as an approach. However, this practice may not be ideally suitable for new Scrum teams.
- Create small tasks:
  - Do not have tasks that span across days together. Its makes it difficult to know the actual progress.



2.2: Agile Methods and Practices - SCRUM

Examples of Scrum Task Board : Example 1

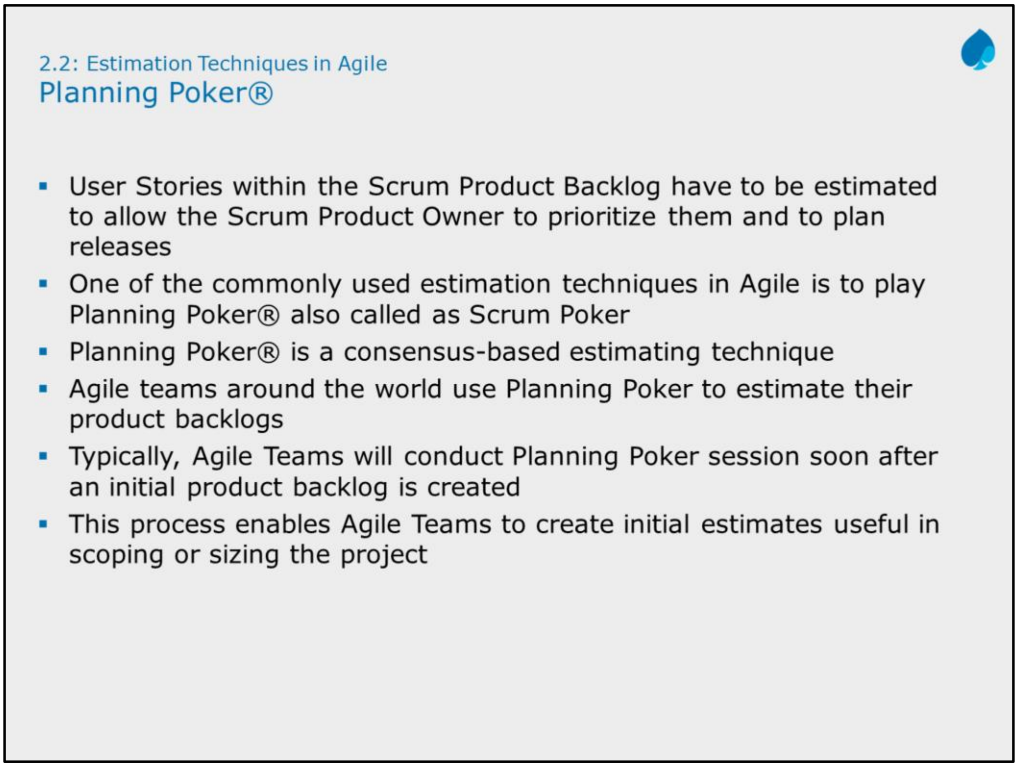
Stories	Not Started	In Progress	Done
Story # 1			Task A Task C Task B
Story # 2	Task A	Task C	Task B
Story # 3	Task B Task D		Task A Task C



2.2: Agile Methods and Practices - SCRUM

Examples of Scrum Task Board : Example 2

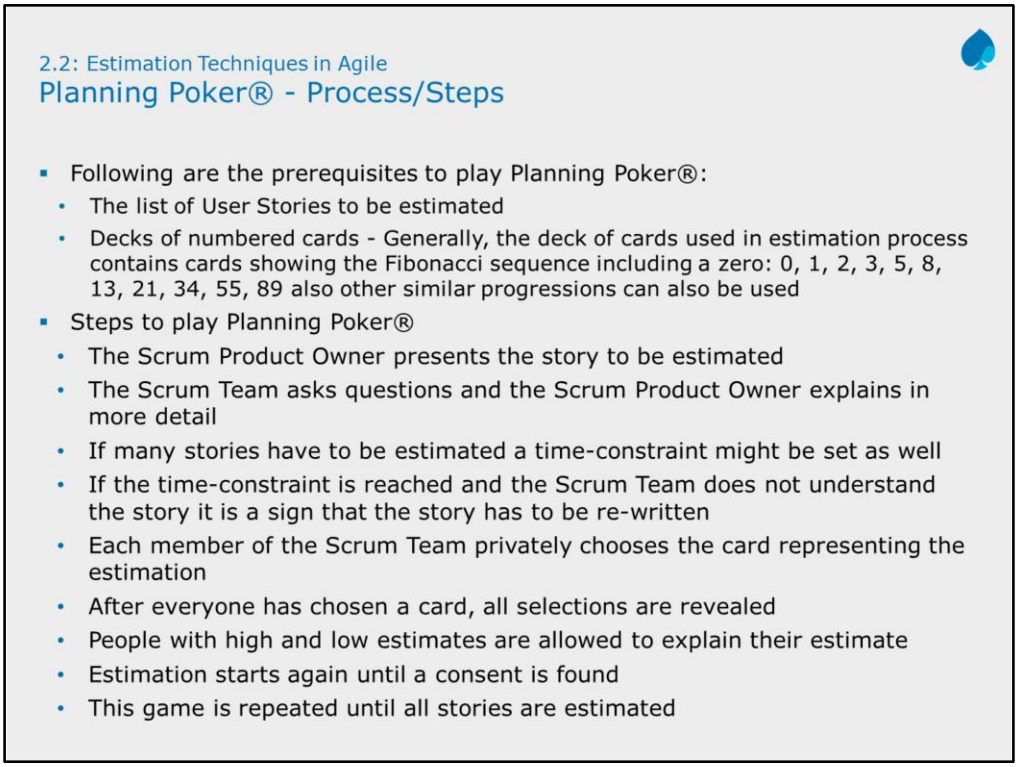
Story	To Do	In Process	To Verify	Done
Story # 1	Task A Task B	Task C	Task D	Task E Task F
Story # 2		Task B		Task C
Story # 3	Task B		Task D	Task A Task C



2.2: Estimation Techniques in Agile

Planning Poker®

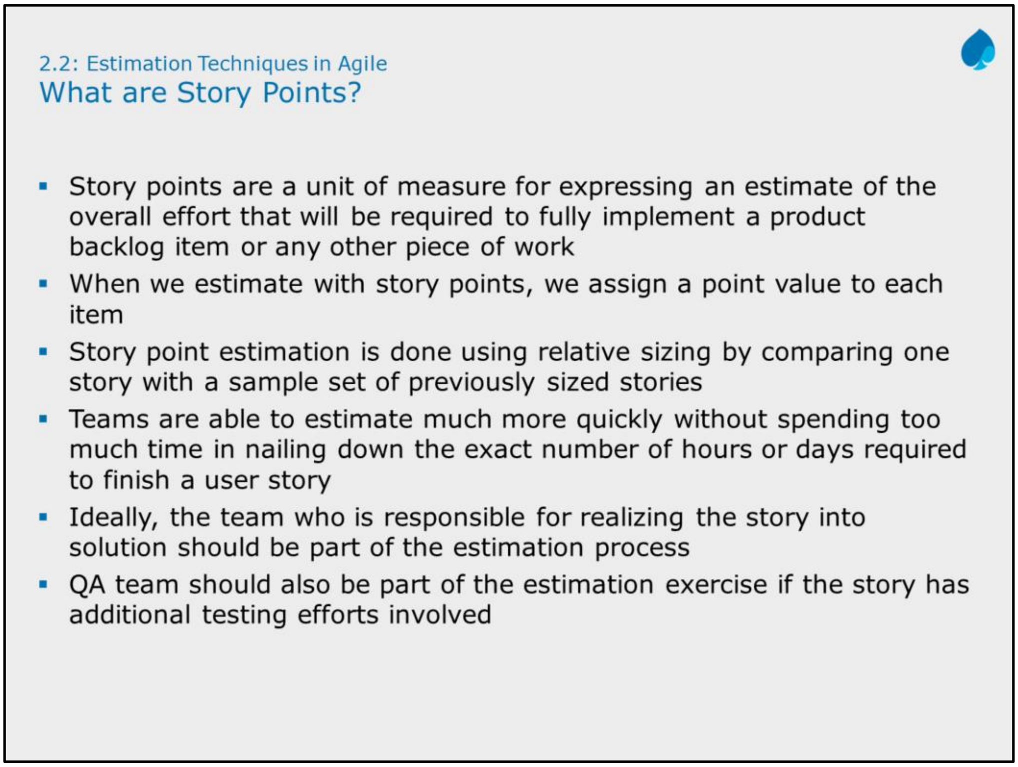
- User Stories within the Scrum Product Backlog have to be estimated to allow the Scrum Product Owner to prioritize them and to plan releases
- One of the commonly used estimation techniques in Agile is to play Planning Poker® also called as Scrum Poker
- Planning Poker® is a consensus-based estimating technique
- Agile teams around the world use Planning Poker to estimate their product backlogs
- Typically, Agile Teams will conduct Planning Poker session soon after an initial product backlog is created
- This process enables Agile Teams to create initial estimates useful in scoping or sizing the project



2.2: Estimation Techniques in Agile

Planning Poker® - Process/Steps

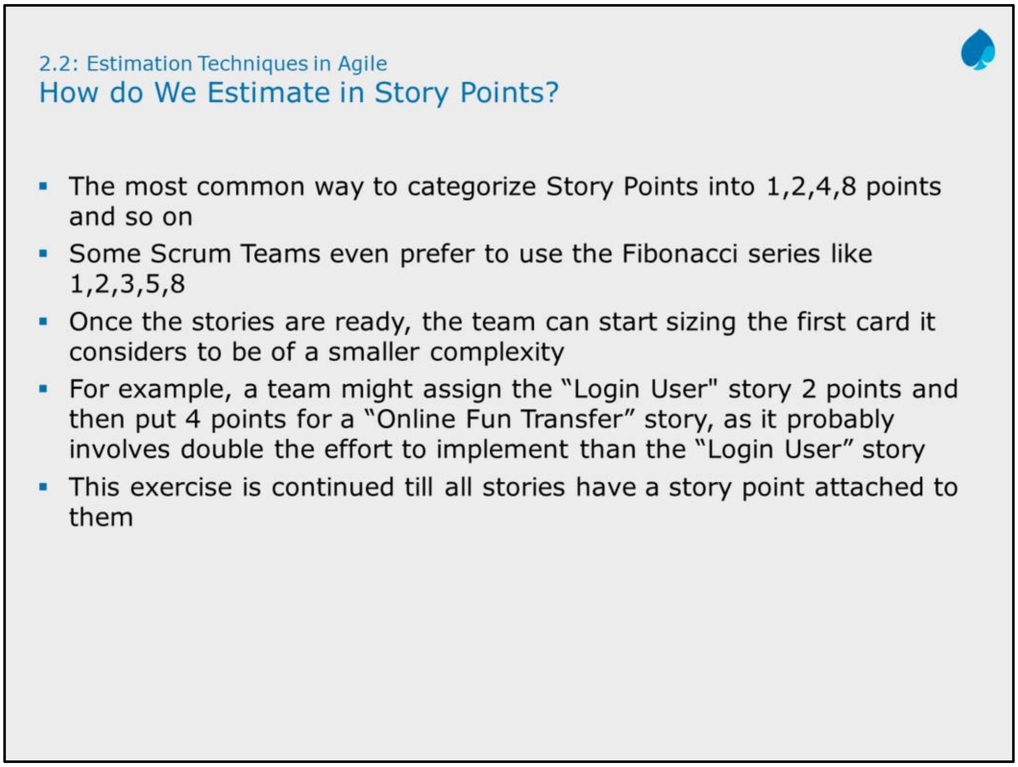
- Following are the prerequisites to play Planning Poker®:
  - The list of User Stories to be estimated
  - Decks of numbered cards - Generally, the deck of cards used in estimation process contains cards showing the Fibonacci sequence including a zero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 also other similar progressions can also be used
- Steps to play Planning Poker®
  - The Scrum Product Owner presents the story to be estimated
  - The Scrum Team asks questions and the Scrum Product Owner explains in more detail
  - If many stories have to be estimated a time-constraint might be set as well
  - If the time-constraint is reached and the Scrum Team does not understand the story it is a sign that the story has to be re-written
  - Each member of the Scrum Team privately chooses the card representing the estimation
  - After everyone has chosen a card, all selections are revealed
  - People with high and low estimates are allowed to explain their estimate
  - Estimation starts again until a consent is found
  - This game is repeated until all stories are estimated



2.2: Estimation Techniques in Agile

What are Story Points?

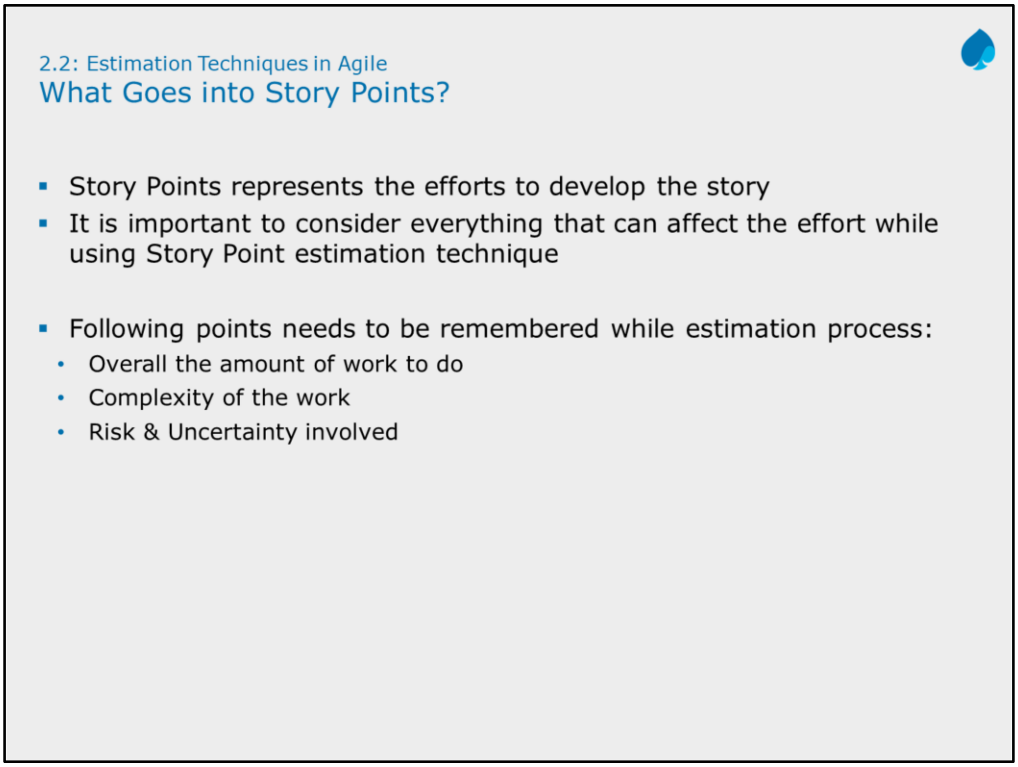
- Story points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work
- When we estimate with story points, we assign a point value to each item
- Story point estimation is done using relative sizing by comparing one story with a sample set of previously sized stories
- Teams are able to estimate much more quickly without spending too much time in nailing down the exact number of hours or days required to finish a user story
- Ideally, the team who is responsible for realizing the story into solution should be part of the estimation process
- QA team should also be part of the estimation exercise if the story has additional testing efforts involved



2.2: Estimation Techniques in Agile

How do We Estimate in Story Points?

- The most common way to categorize Story Points into 1,2,4,8 points and so on
- Some Scrum Teams even prefer to use the Fibonacci series like 1,2,3,5,8
- Once the stories are ready, the team can start sizing the first card it considers to be of a smaller complexity
- For example, a team might assign the “Login User" story 2 points and then put 4 points for a “Online Fun Transfer” story, as it probably involves double the effort to implement than the “Login User” story
- This exercise is continued till all stories have a story point attached to them



2.2: Estimation Techniques in Agile

What Goes into Story Points?

- Story Points represents the efforts to develop the story
- It is important to consider everything that can affect the effort while using Story Point estimation technique

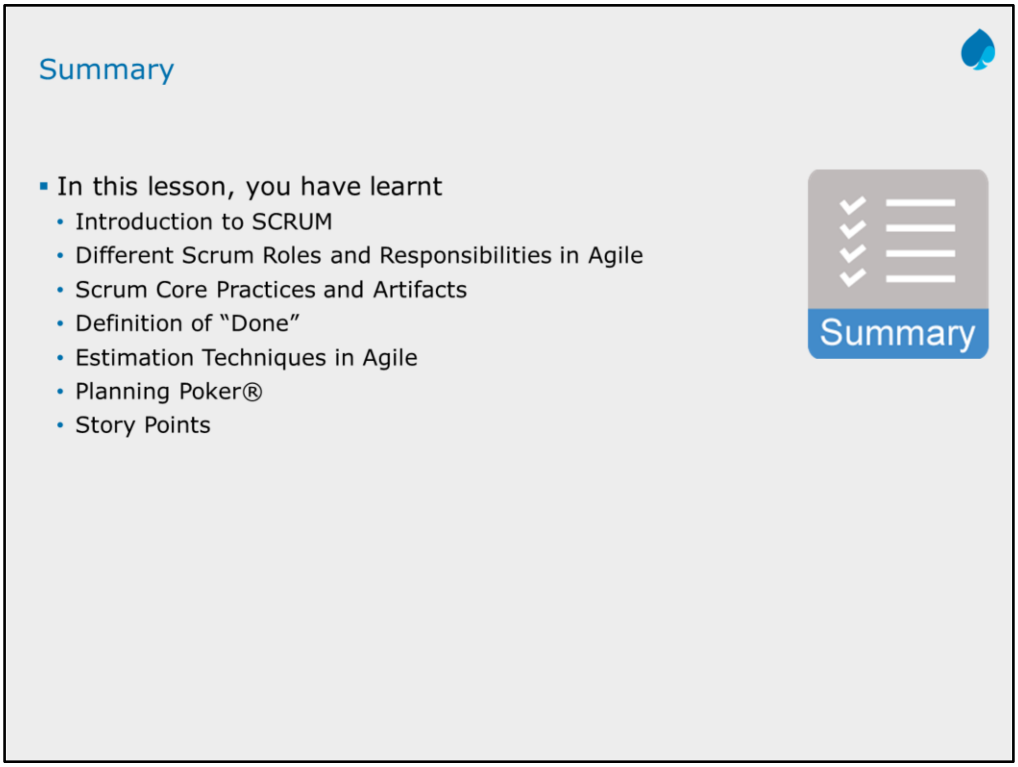
- Following points needs to be remembered while estimation process:
  - Overall the amount of work to do
  - Complexity of the work
  - Risk & Uncertainty involved

Following points needs to be remembered while estimation process:

1. Overall the amount of work to do: Certainly, if there is more to do of something, the estimate of effort should be larger. Consider the case of developing two web pages. The first page has only one field and a label asking to enter a name. The second page has 100 fields to simply be filled with a text. The only difference between these two pages is that there is more to do on the second page. The second page should be given more story points. It probably doesn't get 100 times more points even though there are 100 times as many fields. There are, after all, economies of scale and maybe making the second page is only 2 or 3 or 10 times as much effort as the first page.

2. Complexity of the work: Complexity should also be considered when providing a story point estimate. Think back to the earlier example of developing a web page. Now think about another web page also with 100 fields. But some are date fields with calendar widgets that pop up. Some are formatted text fields like phone numbers or Social Security numbers. This screen also requires interactions between fields. If the user enters a Visa card, a three-digit CVV field is shown. But if the user enters an American Express card, a four-digit CVV field is shown. Even though there are still 100 fields on this screen, these fields are harder to implement. They're more complex. They'll take more time. There's more chance the developer makes a mistake and has to back up and correct it. This additional complexity should be reflected in the estimate provided.

3. Risk & Uncertainty Involved: The amount of risk and uncertainty in a product backlog item should affect the story point estimate given to the item.



Summary

- In this lesson, you have learnt
  - Introduction to SCRUM
  - Different Scrum Roles and Responsibilities in Agile
  - Scrum Core Practices and Artifacts
  - Definition of "Done"
  - Estimation Techniques in Agile
  - Planning Poker®
  - Story Points

Add the notes here.