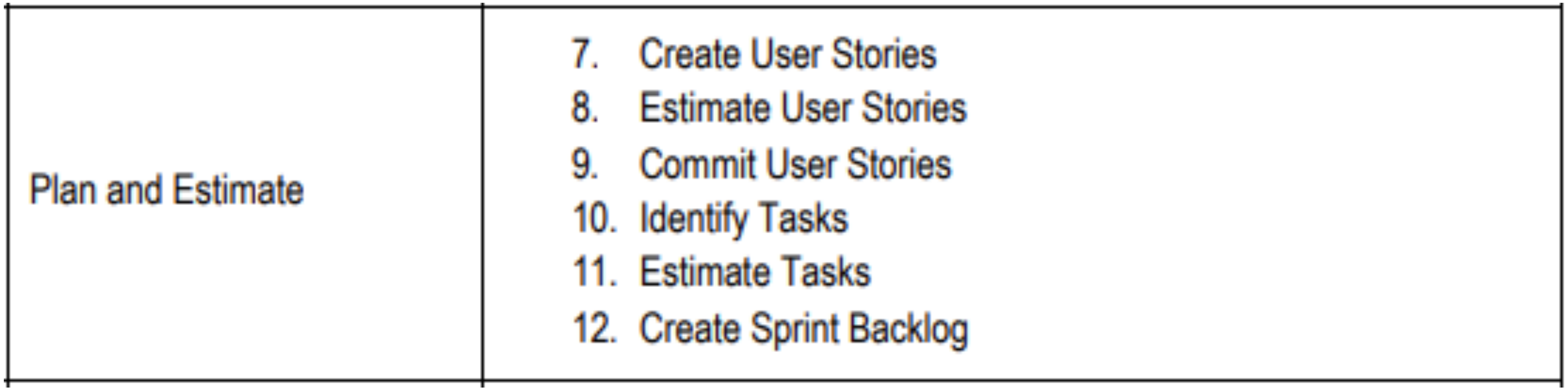
Agile



Plan and Estimate	7. Create User Stories 8. Estimate User Stories 9. Commit User Stories 10. Identify Tasks 11. Estimate Tasks 12. Create Sprint Backlog

Agile

Plan and Estimate phase: Unlike the Initiate phase, which is executed once at the beginning of the project, the planning and estimation phase is executed in all sprints.
The name is self-explanatory and consists of detailing the work to be done in the sprint, its estimation and its planning. This stage also has 6 processes (according to the SBoK@).

Create User Stories: Based on the prioritized Product Backlog (i.e., the team starts with the most relevant and most valuable epics for the customer), the Scrum Core Team begins the process of detailing the epics in User Stories (US).

Agile

Create User Stories

User Stories should be written in the following format:

As a (persona), I want to (perform some action) so that I (can achieve some goal/result/value).

For instance:

Personas Defined in *Develop Epics* (Initiate)

As a **shop-floor operator** I want to be able to **scan the product batch and machine ID being fullfilled** so that **I can trace raw materials batches to finished goods batches**

Agile

In addition to the User Story (US) itself, one of the outputs of the process is the Acceptance Criteria (AC) of this US. As US have a degree of subjectivity, it is essential to include stricter criteria that determine whether or not the requirement can be considered Done at the time of delivery.

A common format is the **Given/When/Then** template that derives from the Behavior-Driven Development (BDD) method, used for writing test cases. Example:

Given I'm a shop-floor operator
When I access the mobile app and go the Replenishment feature
Then the systems requests me to scan the raw material batch barcode and machine ID QR Code, using the device built-in camera
When I scan a raw material batch Or a machine ID different from the one on the production plan
Then the system raises an error
When all scans match the production plan
Then the system records the activity, including my operator ID and the current timestamp

Agile

Concepts to keep in mind:

- The User Story + Acceptance Criteria pair constitute the detail of the specification to be implemented
- Acceptance criteria cannot be changed during the course of the sprint. It's the Scrum Master's job to make sure of that!
- The pairs of User Stories + Acceptance Criteria will be used as test-cases by the QA area, so all the time consumed in this task will be recovered in the testing phase
- Reinforce that US+CA should describe behavior and not technical implementation details… There will be room ahead to think through the details for the technical implementations.
- The entire Scrum Core Team can participate in the design of the US+AC, and the PO is responsible for aligning them with the stakeholders

Agile

Estimate User Stories: Estimating is one of the most controversial and difficult activities to systematize in the software engineering process. Despite the emergence of some techniques, it is still an aspect in which quality is directly related to the experience of the team.
Scrum tries in its design to improve the accuracy of estimates!

There are several methods of estimation, but the SBoK@ points out 4:

- Wideband Delphi
- Planning Poker
- Fist Of Five
- Affinity Estimation

Agile

Something that the estimating team will have to agree on, whatever the method, is the unit of time:

- By **Story Points**. It is a form of estimation by comparison in which a US estimated at 2 points has twice as much complexity as one at 1 point, so it should take twice as long. In the end, the points can be converted into ideal hours or simply the team manages the ability it has in points.
- By **Ideal Hours** in which the team estimates directly in working hours.

Ideal hours is the number of hours required to complete a task without interruptions or unforeseen events (NOT TO BE CONFUSED WITH CALENDAR DURATION):

- Not considering the time of breaks throughout the day
- Without considering time on other tasks, training, vacations, etc
- Without trying to anticipate difficulties, for example in technical mastery
- Not to mention that the person who will execute it is more or less experienced and as such it may take more or less.

Agile

One of the most commonly used ways to estimate the number of hours or points of each US is by using the Fibonacci sequence (or some kind of variant):

$$F_0 = 0, \quad F_1 = 1, \qquad F_n = F_{n-1} + F_{n-2}$$

$$n > 1.$$

1h, 2h, 3h, 5h, 8h, 13h, 21h or
1h, 2h, 3h, 5h, 1d, 2d, 3dh, 5d (variant)

It is not expected that there will be a US of much longer duration... If there is, it means that the team should be divided into smaller units of work

What is the purpose of using a Fibonacci sequence?

- Incorporates the concept of the risk of deviation as HUs become larger and more complex
- It speeds up and facilitates the process: it is easier to estimate whether a US requires 1 or 2 days, instead of 11.5 hours or 12 hours.

Agile

Wideband Delphi. Group Estimation Technique.

- Each element anonymously estimates the US, which is displayed in a plot graph
- The team then discusses the factors that influenced the estimate and a second round of estimates is made
- The process is repeated until all estimates are close and there is a consensus around the value by the entire team

It requires a high degree of maturity from the team to avoid lengthy fruitless micro-detail discussions.

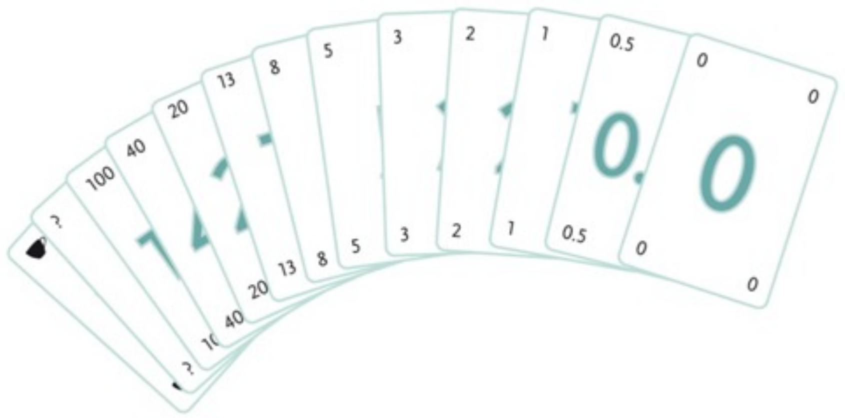
The goal of the first round, being anonymous, is to avoid the trap of groupthink and influence individual thinking (remember what anchoring is?)

Agile

Planning Poker. A derivation of the Delphi Wideband.

Each member of the team is assigned a deck of cards that contain a sequence of points/hours and uses it for the estimation.

- For the US to be estimated, after a brief discussion about it, each element shows the chart with its estimate **simultaneously**.
- Elements who present an estimate above or below the average are invited to explain their motivations.
- The process is repeated until there is consensus and convergence of estimates



Agile

Fist of Five. It is primarily a team voting method, but it can be used for estimation

- For each US, each element uses the number of fingers of the hand to indicate the degree of magnitude and complexity of the hand.
- Similar to Planning Poker, elements with below- or above-average estimates are asked to explain their motivations.
- The process is repeated until there is consensus and convergence of estimates

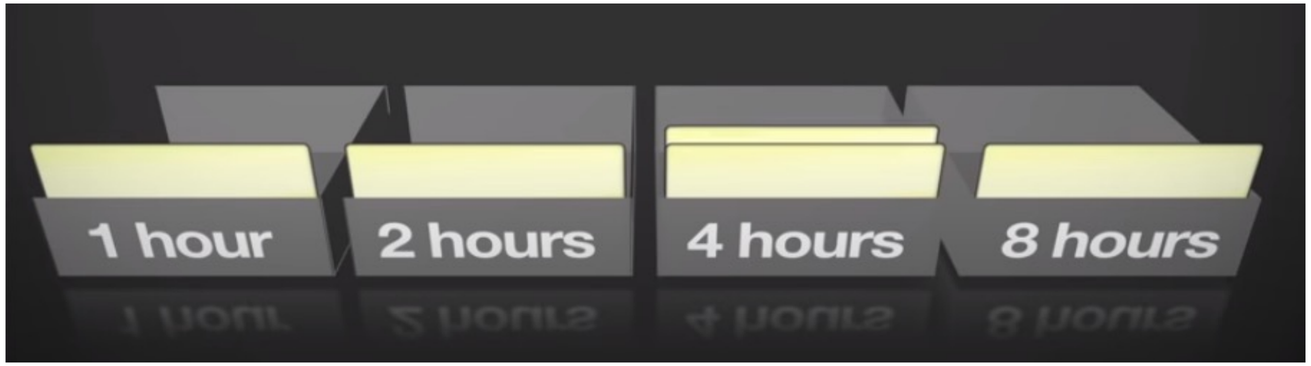
It is an excellent method for comparing the dimensions of the of US, but so that it can be translated into Story Points or ideal hours the team has to find some way to conversion

Agile

Affinity Estimation. Used as a quick way to estimate a large number of HUs, by approximating the magnitude/complexity of the US

Using post-its and a surface containing "boxes" that represent the number of dots/hours:

- Each element receives a subset of the HUs to be estimated and puts them in the "box" that corresponds to their estimate
- At the end of this initial round, in which all the HUs were placed in a "box" by one of the elements, the whole team discusses:
  - Whether the grouped HUs effectively have the same degree of magnitude/complexity
  - Whether there is consensus on the estimate in points/hours



Agile

Conclusions:

- Estimating the software development effort has always been and still is one of the components of the process with the highest uncertainty, ambiguity and risk of deviation
- The methodologies presented in Scrum aim to reduce the risk inherent by:
  - Promote debate and consensus among the entire team
  - Accept that a coarser estimate (Fibonacci sequence) is preferable to spending a lot of time on fine analysis to estimate, which time should be spent generating value
  - Understand that many times at the time of estimation you do not have the desired level of detail and that this should not be a blocker to the process

Agile

Commit User Stories: in this phase, the Scrum Core Team determines and commits to the HUs that will be delivered in the sprint.
To do this, the team must know and use their Sprint Velocity!

Sprint Velocity: is the main metric in Scrum that corresponds to the amount of work the team can produce in a sprint, whether in story points or ideal hours. In experienced teams, this indicator is obtained from previous sprints/projects. Of course, the duration of the sprints has to be comparable.

In short, the Scrum Team, based on its speed and the estimates made from the US, is able to select a group of US for the sprint.

Example: if a team has a velocity of 300 ideal hours in 4-week sprints, it must commit to the US of the prioritized backlog whose sum of estimates is 300 ideal hours.

Agile

If the team does not know its velocity or there have been relevant changes in the team's structure that make it impossible to use historical data, it is always possible to use the following rationale:

- Use the development task as the main driver (as it is the most effort-intensive task in the software engineering process). The remaining tasks, such as QA/Testing, can be a % of the development time. In the case of testing, it is accepted that it corresponds to an effort of 30% of the development value.
- A Scrum Team developer has a daily work schedule of 8 hours
- Assume that only 70% of your time is considered ideal time (the remaining 30% are interruptions such as breaks, emails, phone calls, unforeseen events, etc.), i.e. approximately 5:30h/day.
- Therefore, a developer has a capacity of 27.5h (5.5hx5d) per week and 110h (27.5hx4s) in a 4-week sprint.
- 4 developers therefore have a velocity of 440h (110hx4FTEs) per sprint.

This team of 4 developers can commit to US totaling an effort of 440 hours in ideal hours.

Agile

Identify tasks: This step consists of the decomposition of the HUs into Tasks, which by definition are technical tasks to be implemented:

- Modelling and construction of the BD
- Front-end code
- Back-end code (services, stored procedures and views in the DB etc)
- Web design: Styles + html + images etc
- System Maintenance Scripts
- And so on

At this stage, the Scrum Team should identify the dependencies that can influence the order in which tasks are executed. For example, it is convenient to start the development of a client maintenance after the DB tables involved have been made available.

Dependencies can be external, when they involve tasks that are outside the scope of the Scrum Team. For example, the provision of a component that depends on a purchase process. These tasks carry increased risk because the scrum team has little way to influence. The Scrum Master should be particularly attentive to these dependencies.

Agile

Estimate tasks: Once the tasks have been identified, it is necessary to estimate them in order to allocate the necessary resources for their execution.

It is expected that the sum of the task estimates will be consistent with that of the respective US! However, it turns out to be a counter-proof of the initial estimate of the US.

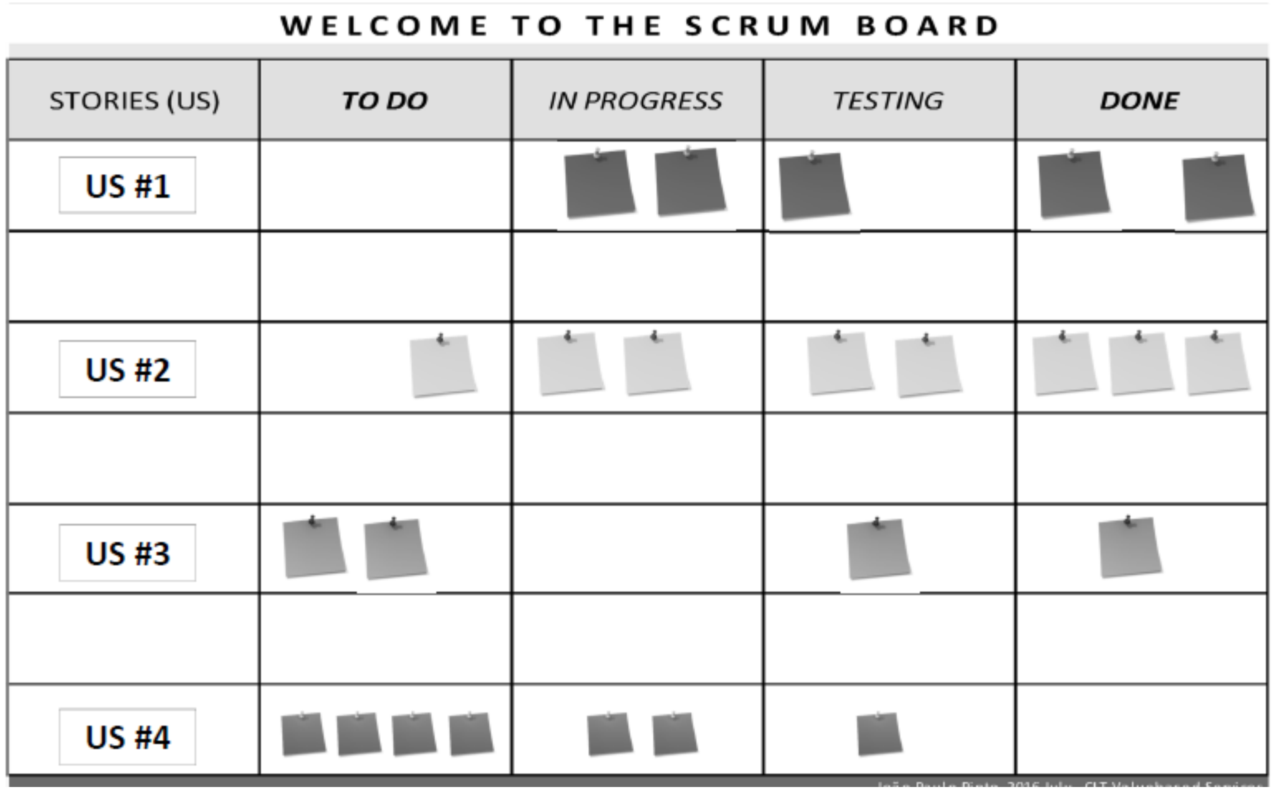
Agile

Create Sprint Backlog: o *sprint backlog* é um subconjunto do *prioritized producto backlog* que corresponde ás US que irão ser implementadas no atual *sprint* resultado de todo o trabalho de planeamento da equipa e com as quais se comprometeu (Commit User Stories)

A forma mais comum de representar o *Sprint Backlog* é através de um Scrum Board

There are many ways to set up a Scrum Board and each organization should design the one that best suits its Scrum implementation. However, as a rule, in a development project there are 4 columns:

- **To-do** – tasks to start
- **In Progress** – development in progress
- **Testing** – full dev and delivered to QA
- **Done** – thorough testing and according to acceptance criteria



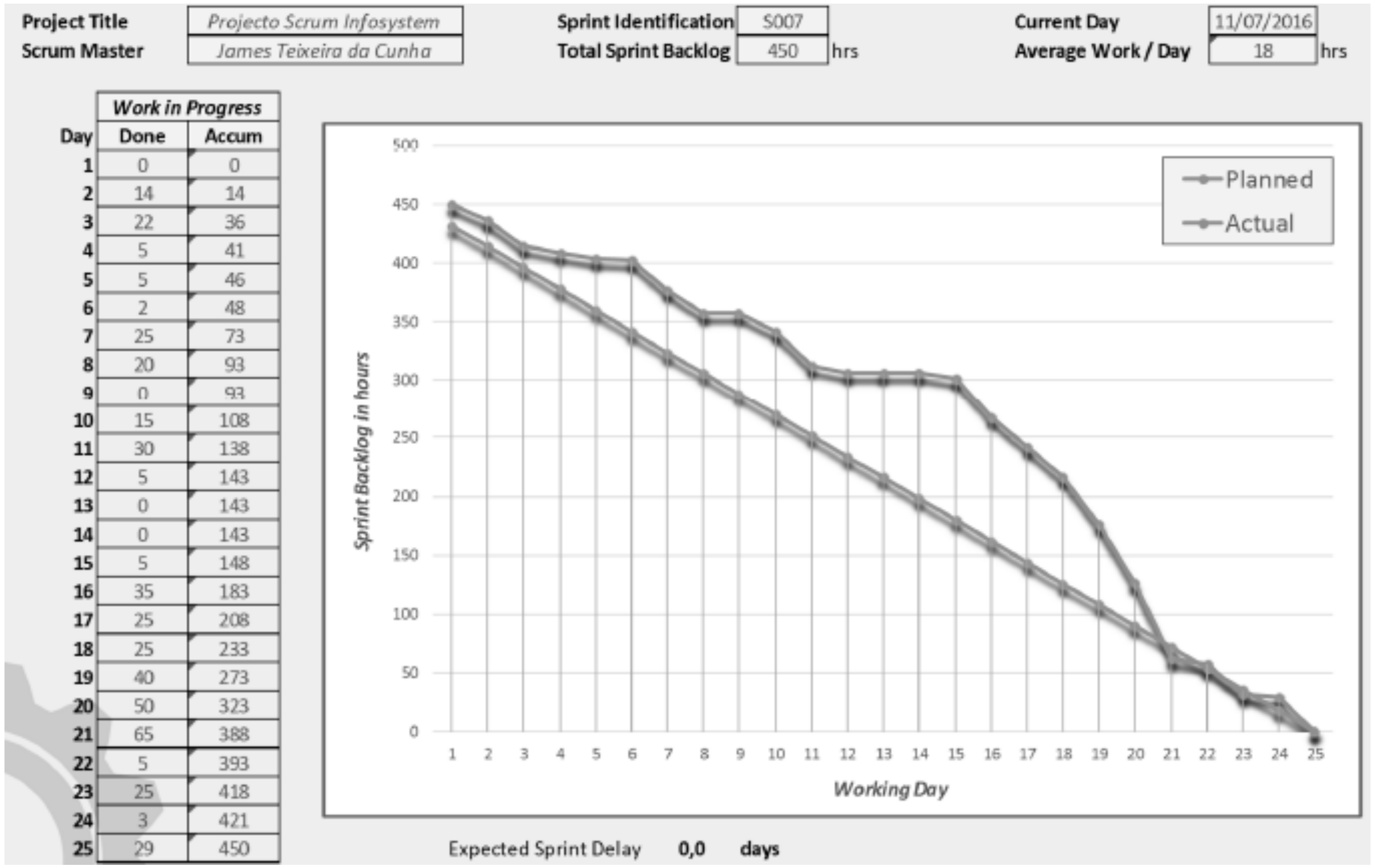
WELCOME TO THE SCRUM BOARD

STORIES (US)	*TO DO*	*IN PROGRESS*	*TESTING*	*DONE*
US #1		■ ■	■	■ ■
US #2	■	■ ■	■ ■	■ ■ ■
US #3	■ ■		■	■
US #4	■ ■ ■ ■	■ ■	■	

Agile

Another key output of this process is the Sprint Burndown Chart, which is a graphical representation of the amount of work still to be done in the current Sprint and what the end date of the sprint will be if the velocity is maintained.

- The Burndown Chart should be updated at the end of each workday, with the work done
- If the Burndown Chart shows that the sprint will not be completed in the estimated time, it is the Scrum Master's job to identify and remove the obstacles



Project Title	*Projecto Scrum Infosystem*	Sprint Identification	S007	
Scrum Master	*James Teixeira da Cunha*	Total Sprint Backlog	450	hrs
Current Day	11/07/2016			
Average Work / Day	18	hrs		

	Work in Progress	
Day	Done	Accum
1	0	0
2	14	14
3	22	36
4	5	41
5	5	46
6	2	48
7	25	73
8	20	93
9	0	93
10	15	108
11	30	138
12	5	143
13	0	143
14	0	143
15	5	148
16	35	183
17	25	208
18	25	233
19	40	273
20	50	323
21	65	388
22	5	393
23	25	418
24	3	421
25	29	450

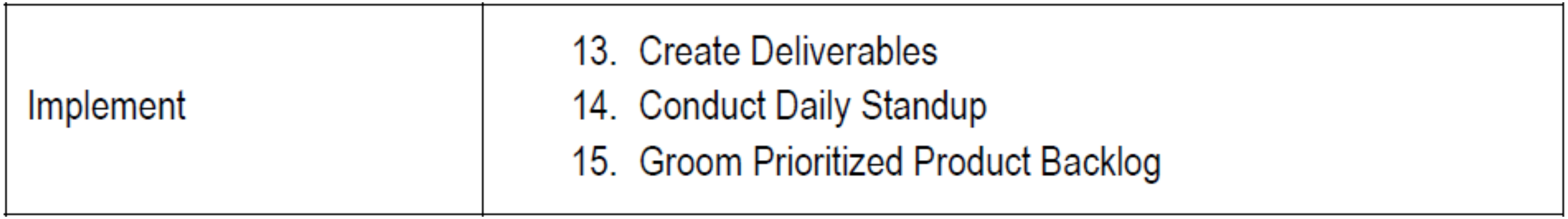
Expected Sprint Delay 0,0 days

Agile

Important Considerations:

- The assignment of tasks to the members of the Scrum Team is not done by either the PO or the SM ... The team is self-organizing, so it distributes the work according to skills, preferences, or other productivity-related criteria
- In the Sprint Backlog, activities should be added in addition to the US of the product backlog, such as actions to mitigate the risks detected
- Once the Sprint Backlog is closed, no new US can be added. it is the role of the Scrum Master to ensure this discipline
- However, if there is a need for change (AND DON'T FORGET THAT CHANGE IS ONE OF THE ASPECTS OF SCRUM), the new US should be added to the prioritized backlog for future sprints

Agile



Implement	13. Create Deliverables 14. Conduct Daily Standup 15. Groom Prioritized Product Backlog

Agile

Phase **Implement:** is the hands-on phase of the sprint in which the Scrum Team executes the development, testing, design, etc. tasks previously agreed upon in the planning phase.

Create Deliverables:

- The Scrum Team executes the necessary developments
- The anomalies (bugs) that the QA team detects in the tests should be recorded in some kind of bug tracking platform, not only for control but also for quality metrics.
- Remember that the Scrum Team is self-organizing, so its work is not managed or coordinated by the Scrum Master and/or Product Owner
- At the end of each day, the Scrum Board should be updated:
  - Cards moved accordingly
  - And the burned-time reported for the purpose of updating the Burn Down Chart
- An impediment log (list of blocks and obstacles to the success of the sprint) should be kept for discussion and resolution

Important note: the Scrum Board should be updated throughout the day or at the latest at the end of the day. It is highly unproductive to consume daily stand-up time to update these indicators.

Agile

Conduct Daily Standup: one of the most important and well-known Scrum ceremonies. It consists of a daily team meeting and is the maximum moment of collaboration between the Scrum Core Team:

- It occurs daily and should have a maximum duration of 15min (time-boxed)
  - There are no postponements or extensions.
  - It should take place at the beginning of the day and start at the time, with whoever is present!
- It is essential that the Scrumboard and the Burndown Chart are up to date
- Mandatory participation of all elements of the Scrum Team and the Scrum Master
- The Product Owner may participate if clarification is needed
- The daily should take place in front of the Scrum Board
- The role of the Scrum Master is to be a facilitator, so he must help the Scrum Team to overcome the reported difficulties

Agile

- In the daily each element gives a note on 3 aspects:
  - What I Completed Yesterday
  - What I'm going to conclude today
  - I'm running into an impediment or obstacle (impediment log)
- <u>Very important: </u>the time of the daily should not be used to discuss how to overcome obstacles. This discussion should take place after the daily with the strictly necessary elements.

Agile

Groom prioritized backlog: grooming means "to refine", "update", "tidy up" and in Scrum it is the ultimate representation of the ability to react to change and continuous improvement:

- Product backlog grooming is not a ceremony… It's something that takes place throughout the sprint
- It is the responsibility of the Product Owner
- It consists of:
    - Remove Epics/User Stories That Have Lost Value
    - Add new User Stories as a result of the feedback given by the client in the validation of previous sprints
    - Add Epics/User Stories, as a result of context changes: legal, regulatory, market, lessons learned
    - Add tasks/actions to mitigate risks that are identified
    - Re-ordering User Stories (reprioritization)
    - **Fundamental:** grooming is done to the prioritized product backlog and not to the Sprint Backlog… means that the impact will be on future sprints and not the current one

Agile

Review and Retrospect	16. Demonstrate and Validate Sprint 17. Retrospect Sprint

Agile

Phase **Review and Retrospect:** final phase of the sprint in which the deliverables of the sprint are shown and validated and there is also a retrospective qualitative analysis to obtain lessons learned

Demonstrate and validate Sprint:

- The Scrum Team presents or demonstrates deliverables to the PO and/or relevant stakeholders, in a formal ceremony: Sprint Review Meeting
- The PO validates the deliverables according to the Acceptance Criteria of each User Story in turn:
  - Can be accepted -> are marked as "Done"
  - Can be rejected - > remain in the prioritized product backlog to be addressed again in future sprints
- Remember the fundamental role of the Scrum Master throughout the sprint so that the requirements/acceptance criteria were not changed during the sprint

A sprint can result in a (sub-)product that can be delivered for immediate use! Hence the concept of Scrum's gradual and incremental deliveries.

Agile

Retrospect Sprint: As in any continuous improvement cycle, it is necessary to understand what went well to repeat and what went less well in order to learn and prevent future mistakes. At the end of each sprint, it is necessary to do a retrospective analysis:

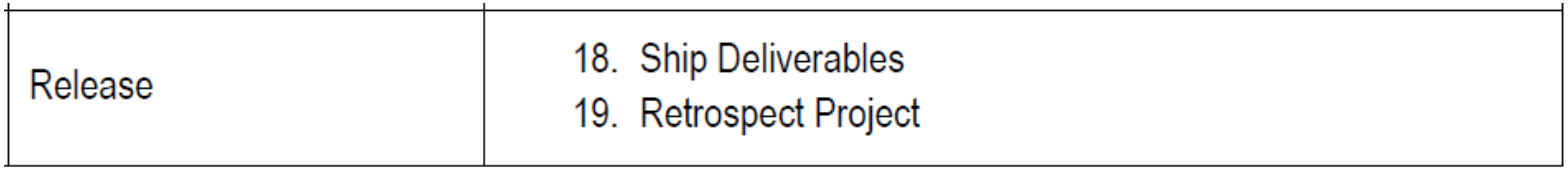
- There is a ceremony for this retrospective session: Retrospect Sprint Meeting
- The Scrum Team and the Scrum Master must participate. The presence of the Product Owner is optional, but advisable if it makes sense
- The goal is to follow a **KBS** logic**:**
    - Things the team needs to **Keep** doing: best practices
    - Things the team needs to **Begin** doing: process improvements
    - Things the team needs to **Stop** doing: process problems and bottlenecks
- It is necessary to create a relaxed environment to avoid turning into a "washing of dirty laundry". That is, it should be focused on identifying root causes and improvement actions and not on blame and those responsible.

Agile

Metrics: The Restrospective Meeting is a moment of excellence to present some metrics and indicators to the team and even compare with previous sprints:

- **Team Velocity:** the number of points or ideal hours run in the sprint
- **Done rate:** % of User Stories accepted (Done), of those that were committed by the team
- **Review feedback ratings:** implement a rating mechanism by stakeholders and POs at the Sprint Review Meeting that qualitatively evaluates the deliverables of the sprint
- **Product Quality:**
  - Escaped defects: Number of anomalies (bugs) detected by users in production. Applicable in situations where deliverables from previous sprints have been deployed. A KPI for the QA team.
  - Defect Density: number of anomalies (bugs) detected by QA on the number of points or ideal hours of the sprint
  - Test coverage: % of code covered by automated testing
- And others…

Agile



Release	18. Ship Deliverables 19. Retrospect Project

Agile

Phase **Release:** corresponds to the closing phase of the project. No surprises are expected, because during the course of the various sprints there were several moments of feedback and respective adaptation (at least theoretically!).

Ship deliverables:

- If there were releases in previous sprints, this will be one more and potentially the last one before entering the follow-up or even support period, RTM, Release to Manufacturing.
- An official sign-off of the project by the client is highly advisable, with a view to acknowledging the final delivery. In Scrum this sign-off is called the Working Deliverables Agreement

Agile

Retrospect project: Similar to the Retrospective Sprint, but at the project level. It is often referred to as a project post-mortem meeting.

It differs from the retrospective sprint in that:

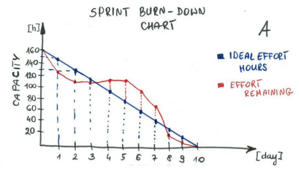
- Senior management and stakeholders may participate
- Suggestions for changes to the Scrum Guidance Body Recommendations may be made, in light of the lessons learned

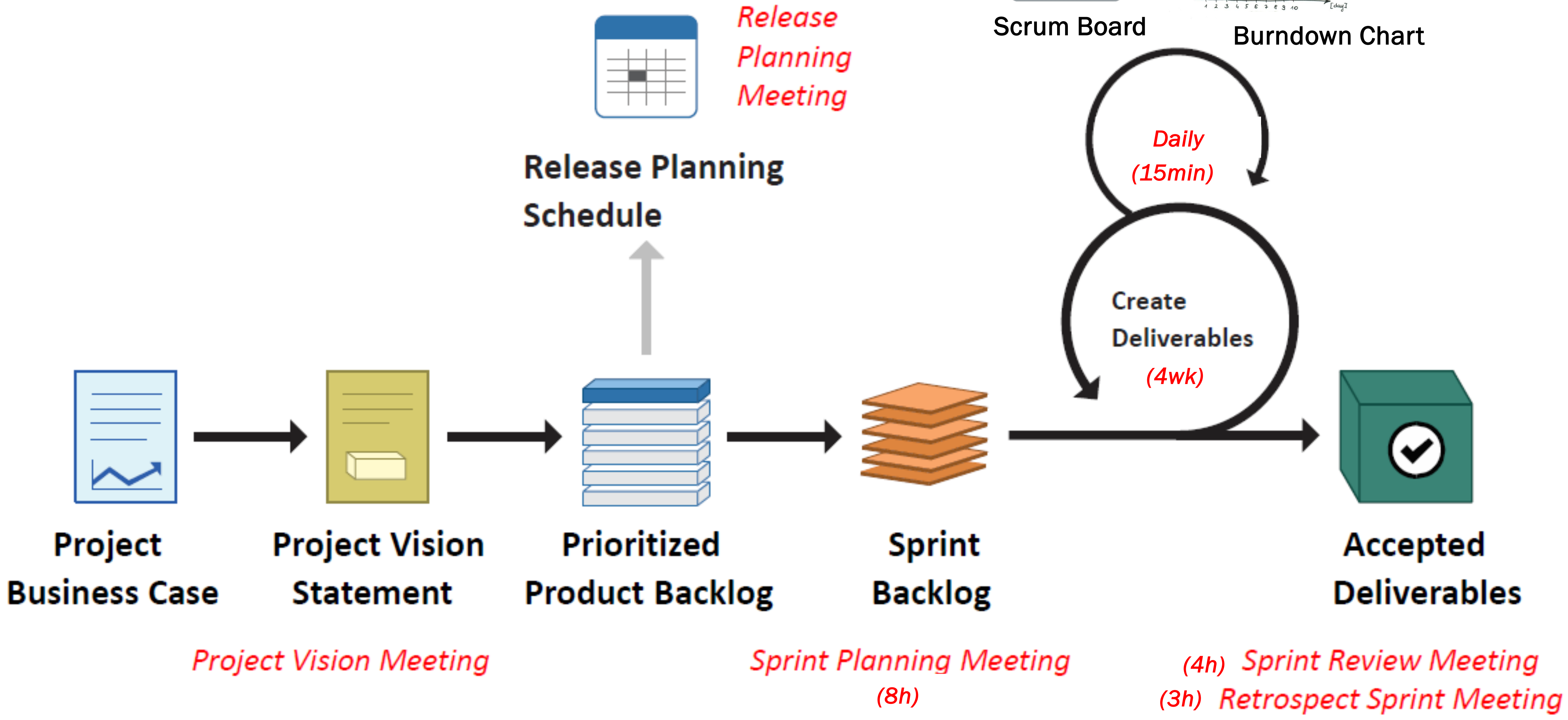
If the above two conditions are not met, it may not provide more value than the retrospective sprint has already done...

Agile

Conclusion: We have seen 6 principles, 5 aspects, and 19 processes of the Scrum Framework, which can convey a message of complexity. However, Scrum can be implemented in an **agile** and **simplified** way itself

The relevant ceremonies and artefacts





SCRUM ON A PAGE

SCRUM PRINCIPLES

Empirical Process Control

Scrum believes in making decisions based on observation and experimentation rather than detailed upfront planning.

Self-Organization

As opposed to the traditional command and control style of management, Scrum believes that the knowledge workers of today have lot more to offer than just their technical expertise and deliver greater value when self-organized.

Collaboration

Scrum believes that product development is a shared value creation process that needs all the stakeholders working and interacting together to deliver greatest value.

Prioritization

Delivering the greatest value in the shortest amount of time requires prioritization and selection of what will be done from what needs to be done.

Time-boxing

Time is treated as a limiting constraint and time-boxing is used as the rhythm to which all work and contribute to.

ROLES

CORE:

Product Owner

- Responsible for assessing viability and ensuring delivery of the product
- Decides on the product vision, release dates and is the voice of the customer
- Prioritizes items in the Product backlog according to business value
- Ensures transparency and clarity on the Product Backlog items
- Provides acceptance criteria and Inspects deliverable to validate them

Scrum Master

- Acts as a guardian for the team and protects them from external interference
- Does not get work done but only facilitates the team and removes their obstacles
- Ensures the team follow and implement Scrum practices
- Acts as a motivator and a coach to the team
- Acts as a change agent, ensuring smooth and effective change process

Scrum Team

- Typically a small team of 6-10 members with no further sub-division of teams
- Cross-functional and self-organizing and enjoys complete autonomy during a sprint
- Members are generalists across domains and Specialists in at least one area
- Equality maintained among all members of the team
- Responsibility of the work lies with the whole team

ANCILLARY:

Stakeholders

- Observe
- Support
- Advice

ARTIFACTS

Product Vision

Proposed by the Product Owner and accepted by the team, it is a one sentence aim for the product.

Product Backlog

A list of requirements that, when turned into potentially shippable product functionality, will deliver the product vision. Owned and prioritized by the Product Owner but may be appended to by anyone.

Sprint Goal

Proposed by the Product Owner and accepted by the team, it is a one sentence aim for the current sprint.

Sprint Backlog

The Sprint Backlog is a subset of the Product Backlog that a team commits to in a particular sprint. It is a list of decomposed and estimated tasks that only the team can modify.

Blocks List

List of obstacles and pending decisions faced by the team and maintained by the Scrum Master.

Product Increment

The potentially shippable deliverable of the team at the end of each sprint that satisfies the acceptance criteria and is done.

MEETINGS

Kickoff Meeting

Scrum projects start with a Kick-off meeting where the product vision is decided upon by the Product Owner and the Stakeholders funding the project.

Release Planning

The Release Planning meeting is used to decide on the duration required for iterations and meetings of a sprint cycle as well as a release plan for multiple sprint cycles.

Sprint Planning

Each sprint begins with a planning meeting that is time-boxed to eight hours for a one-month sprint. The Sprint Planning meeting accomplishes Objective Definition & Tasks Estimation.

Daily Scrum

The Daily Scrum meeting is a short time-boxed meeting (generally 15 minutes) for the Team members. Each Team member answers the following questions:

1) What did I do yesterday?
2) What will I do today?
3) What's in my way?

Sprint Review

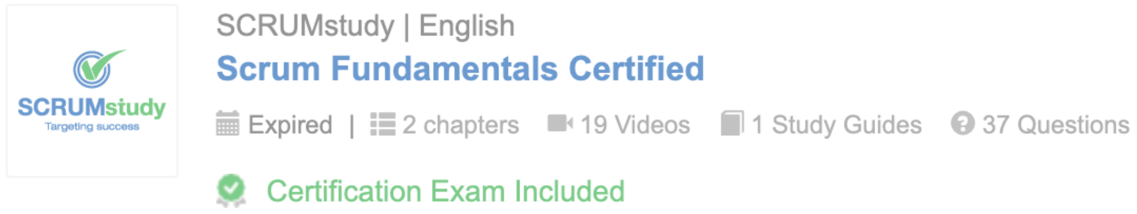
Sprint Review is for the Team to present the end-deliverable of the sprint to the Product Owner and Stakeholders. The done backlog items can either be accepted or rejected by the Product Owner.

Sprint Retrospective

The Sprint Retrospective is an opportunity for the Team to inspect the previous sprint and identify potential improvements that can be implemented in future sprints.

Suggestion - **Scrum Fundamentals Certified**

https://www.scrumstudy.com/


SCRUMstudy | English

Scrum Fundamentals Certified

Expired | 2 chapters 19 Videos 1 Study Guides 37 Questions

Certification Exam Included