DOCUMENT NO. 17019 | VERSION 1.7

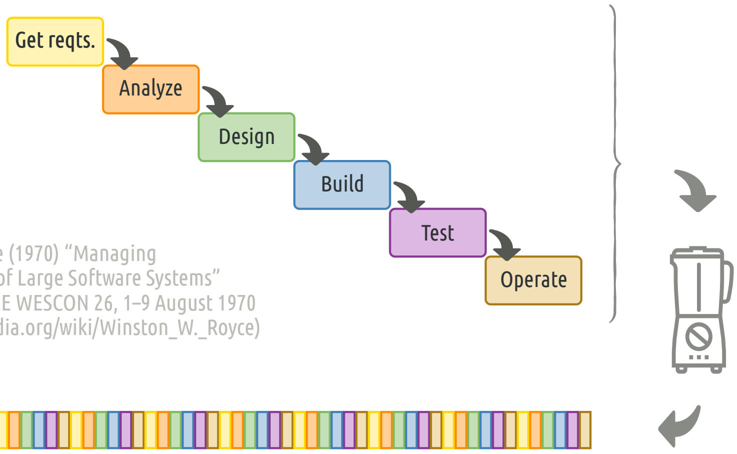
openSE, PM Agility and SCRUM
Part 6

Table of Content

0. Foundations

What is SCRUM ?

An alternative to the traditional project management "**waterfall model**" theorized in the 1970's*



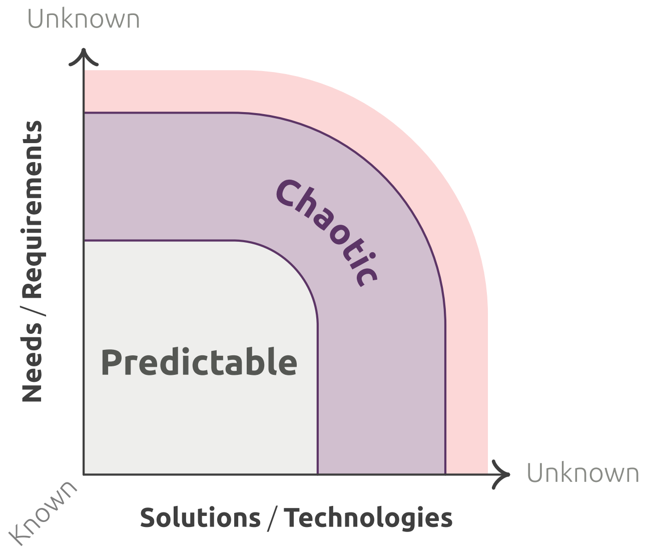
*Winston W. Royce (1970) "Managing the Development of Large Software Systems" Proceedings of IEEE WESCON 26, 1–9 August 1970 (https://en.wikipedia.org/wiki/Winston_W._Royce)

SCRUM :

many fixed-duration iterations referred to as **sprints**

Pierre Bonnal

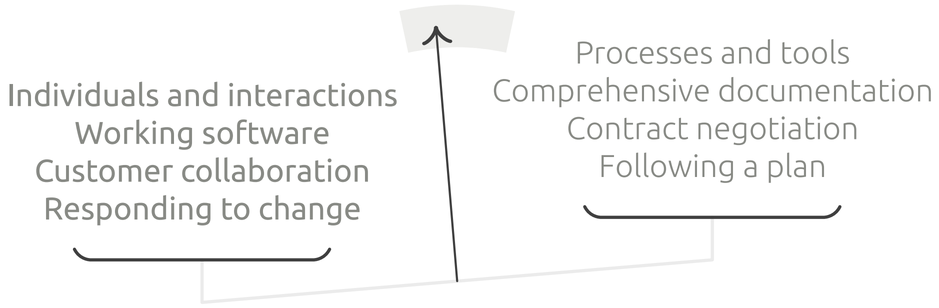
What is SCRUM ?



What is SCRUM ?

Manifesto for Agile Software Development*

> 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
Working software
Customer collaboration
Responding to change

Processes and tools
Comprehensive documentation
Contract negotiation
Following a plan

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

*AgileManifesto.org

Pierre Bonnal

What is Scrum ?

Theoretical background

- Grounded on the empirical process control theory → **Empiricism***
  1. Knowledge comes only or primarily from sensory experience
  2. Decisions shall be made based on this knowledge
- Relies on **three pilars**, namely:

Transparency — e.g. sharing a common language

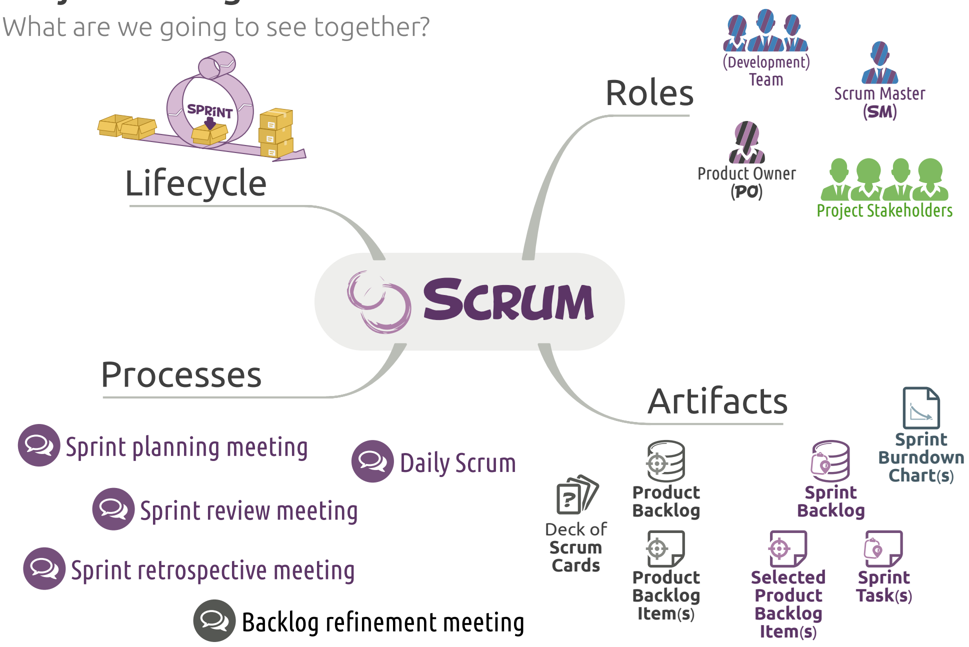
Inspection — e.g. frequent reviews

Adaptation — e.g. "along-the-way" strategy

*en.wikipedia.org/wiki/empiricism

Project Management

What are we going to see together?



Scrum

Lifecycle

Roles

- (Development) Team
- Scrum Master (SM)
- Product Owner (PO)
- Project Stakeholders

Processes

- Sprint planning meeting
- Daily Scrum
- Sprint review meeting
- Sprint retrospective meeting
- Backlog refinement meeting

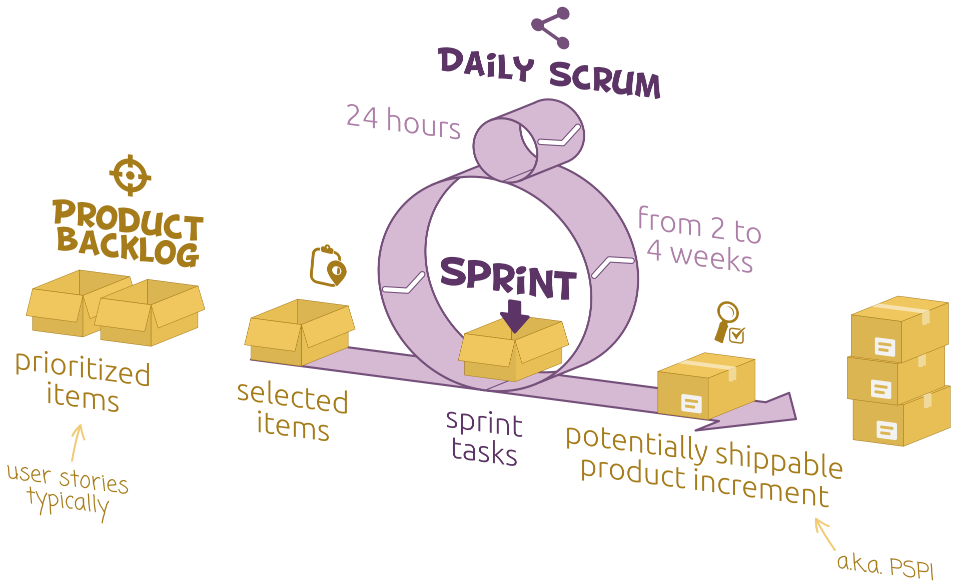
Artifacts

- Deck of Scrum Cards
- Product Backlog
- Product Backlog Item(s)
- Sprint Backlog
- Sprint Burndown Chart(s)
- Selected Product Backlog Item(s)
- Sprint Task(s)

1. Lifecycle

What is SCRUM ?

Its lifecycle



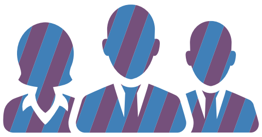
2. Roles

Roles

Product Owner (PO)

- Is necessarilly a **single individual**
- Is responsible for the “**return on investment**” (ROI), so has authority on **prioritization** of product backlog items
- Conveys all **requirements** from stakeholders
- Is the **final arbiter** w.r.t. requirement questions, however, doesn't necessarilly knows all the requirements!
- Owns the **vision** w.r.t. the project deliverable
- Makes the **business decisions**
- Is more “**what-focussed**” than “how-focussed”

Roles

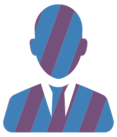


(Development) Team

- Is a **cross-functional team** of 4 to 9 people
- Is focussed on delivering "**potentially shippable product increment**" at every sprint
- Is organized as a **collaborating team**, i.e. is self-organized, promotes natural leadership

  collaboration (≠ co-operation)
- Works in a **team room**, and uses intensively **information sharing tools**

Roles



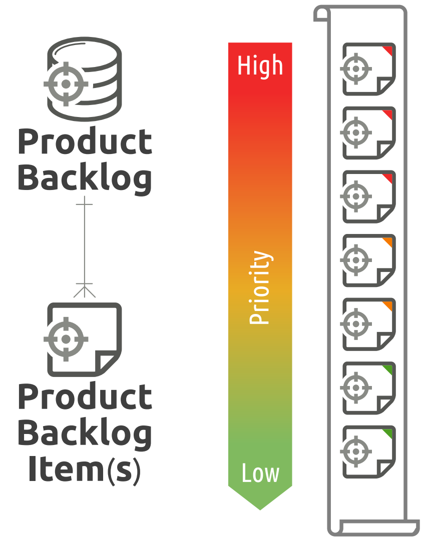
Scrum Master (SM)

- Is also necessarilly a **single individual**
- Has **no management authority**, and for this reason, cannot be a line manager nor a project manager
- Is a **facilitator** who protects the development team from distractions and interruptions
- Can facilitate **one team** at a time
- Teaches project participants **how to use Scrum**
- Promotes **improved engineering practices**
- Convenes meetings and enforces ***timeboxes*** (timekeeper in meetings)

3. Artifacts

Artifacts

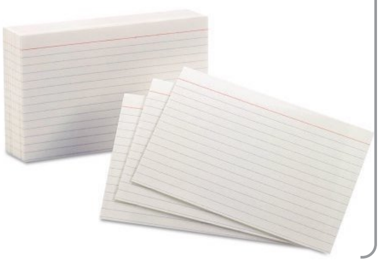
Product Backlog



- Is a prioritized and dynamic list of **product backlog items**; typically a list of **user stories**
- Lists all **features**, **functions**, **requirements**, **enhancements**, and **fixes** related to the product
- ⚠ Is not a list of activities or of tasks!
- Is owned, ordered (prioritized) and maintained by the **product owner**
- Is populated by **anyone** (PO, SM, team members, and stakeholders)
- Evolves as the project progresses (items can be added, removed, modified, morphed, etc.)

User Stories

- Are informal, short and high level descriptions of **requirements** or **features**, containing just enough information so that teams can produce a reasonable estimate of the effort required to implement each of them
- Facilitates **sensemaking** – i.e. the gathering of meaningfull collective experience – and **communication** among stakeholders
- Help shift the focus from writing about the requirements to talking about them → concept of "**C**onversation"
- Are written/owned by the product owner, although they can be initiated by everyone
- Practically, are traced by means of **C**ards (index cards or sticky notes)



User Stories (continued)

a.k.a. labels or titles

- Are formulated by means of **user story statements**:

As a ⟨role⟩, I want ⟨something⟩ so that ⟨benefit⟩

type of user — goal, capability — reason, rationale

- Are prioritized, i.e. ordered; therefore feature a **priority index**, e.g.:
  - Three-level scale such as: high, medium, low priorities
  - MoSCoW scale: must, should, could, won't
  - 1-10 scale
- Are estimated, typically in **story points**, that is a kind of relative scale
- Feature **conditions of satisfaction** → **C**onfirmations that ensure that the objectives of the story have been reached
- Can either be **not done** or **done** (not in between, e.g. partially done)
- Can be grouped in **epics** and **themes**

Epics

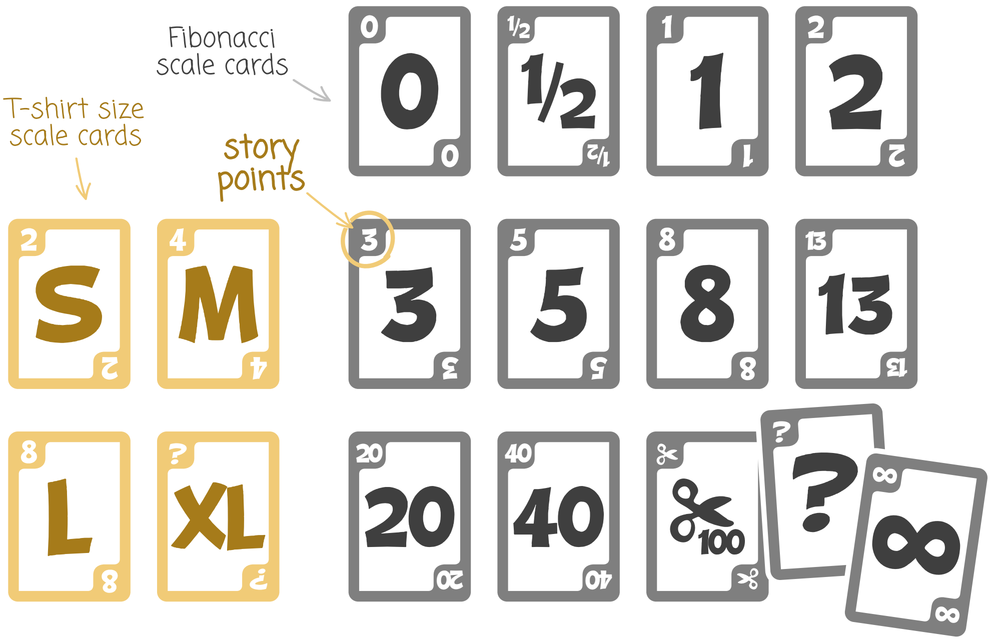
- Are large bunches of work containing **several user stories**
- Are **too big** to be implemented in a single sprint; therefore, they need to be disaggregated into smaller/manageable user stories at some point
- Are necessarily **lower-priority** user stories

Themes

- Are grouping of user stories and epics adressing a same family of features

Scrum Estimation Cards

a.k.a. Planning Poker® or Scrum Poker or Scrum Cards



Story Point

Definition

- A story point is a **measure of magnitude**
  It's a way to understand the **relative level of effort** of a specific user story as compared to other user stories of the product backlog
- Story points enable effort to be estimated without trying to estimate how long it will take: **story points ≠ ideal person·hours or person·days**

To derive an estimate for the **duration** of a project or a sprint: divide the story points for the user stories by the velocity of the team, given by the number of story points achieved over the last sprint(s):

$$\text{duration}_j = \frac{\sum_i \text{story points}}{\text{team velocity}}$$

Ideal (**person·**)**hour** — An hour of work where the project participant solely focuses on the task at hand without any interruptions like phone calls, electronic mails, or chat messages. A general agreement sets an average default capacity of six ideal hours per day (for an eight hour working day) for every team member. That is a 75% effectiveness.

Story Point

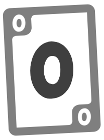
Why use the story point instead of person·hours or person·days?

1. Business stakeholders have a difficult time understanding the concept of “ideal person·hours” or “ideal person·days”
   They tend to forget the “ideal” part!
2. Disparity between how fast certain team members can complete work
   An experienced team member who has worked on similar projects for years will think, "I can do that in about an hour", while the junior one might take more than one day to accomplish the same task
3. High degree of optimism present with most highly creative teams
   It is easy to think “It’s a day,” while it's really three days’ worth of work

Story points eliminate these problems by disconnecting the size of a user story from the time it will take to implement it

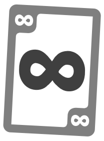
When time is taken off the table, teams tend to be much more effective at estimating relative size

Scrum Cards

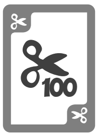


The zero card — Sometimes the list of stories will include one that is very straightforward, where the team believes it will only be a few minutes of work, or perhaps an hour or two.
The story is perceived too small to be an ½ story point value effort!

The question mark card — The team should discuss each story before voting, but sometimes a team member really has no idea what to estimate.
In this case the team should discuss the story further and vote again.
As a general rule, if this card is used it means the team needs to discuss the stories more so every team member understands them enough to be able to vote with a story point value.



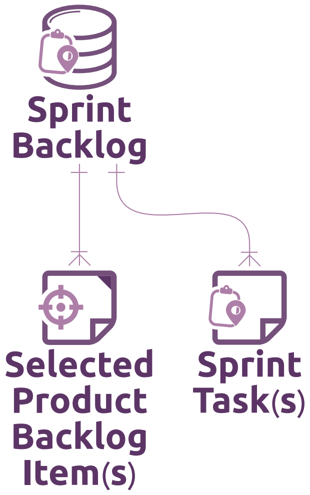
The infinity card — Sometimes stories are so large (epic), insufficiently defined or risky that the team member doesn't feel comfortable placing any sort of story point value on them.
These types of stories either need clarification or need to be scaled down into smaller pieces.



The one-hundred card — Sometimes the team member just sees that the story is an epic and needs to be disaggregated into more atomic stories.

Artifacts

Sprint Backlog & Board

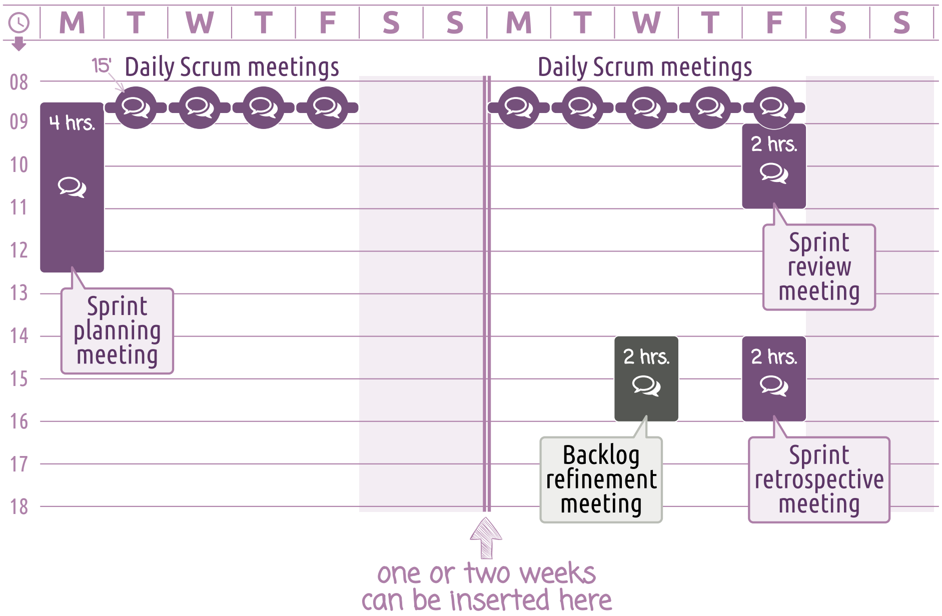


- Is a list made of **selected product backlog items** to be developed during the sprint
- Is augmented with the **set of tasks** required to achieve the sprint goal, i.e. demonstrating a PSPI at the end of the sprint

4. Meetings (Events)

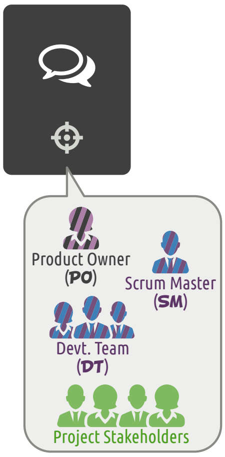
Processes

Sprint Meetings



Processes

Meetings



a.k.a.
Backlog Grooming Meeting
Backlog Estimation Meeting
Backlog Story Time Meeting

Backlog Refinement Meeting

- On **day** ***n*** – 2 p.m. (2 hours)
  and at least once before the project starts
- Aims at **looking at the product backlog**, i.e.:
  - Clarifying items and requirements
  - Estimating the effort → Scrum Cards
  - Decomposing epics into smaller items
- Each product backlog item shall be:

Independent
Negotiable
Valuable
Estimable
Small
Testable

} "INVEST"

"INVEST"

Independent — As the team learns and implements modern engineering practices that reduce item dependencies, the product owner gains the ability to set priorities by business value rather than technical considerations.

Negotiable … and Negotiated — A good product backlog item shall be negotiable and not an explicit contract for features. Rather, details will be co-created by the product owner and the team along its implementation.

Valuable — A well-formed product backlog item shall be stakeholder- or user-centric. So its business value is clear.

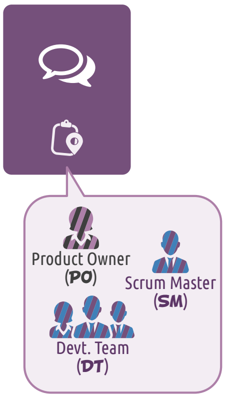
Estimable — If project participants cannot set a rough estimate of the effort required for implementng a product backlog item, this means that it is insufficiently clear and shall be discussed further.

Small — A single product backlog item shall be no bigger than a quarter of the team effort for one sprint. Ideally, it should not exceed two team·days.

Testable — Each product backlog item shall have a clear and bright finish line agreed between the product owner and the team.

Processes

Meetings



Sprint Planning Meeting

- On **day 1** a.m. (4 hours)
- Aims at:

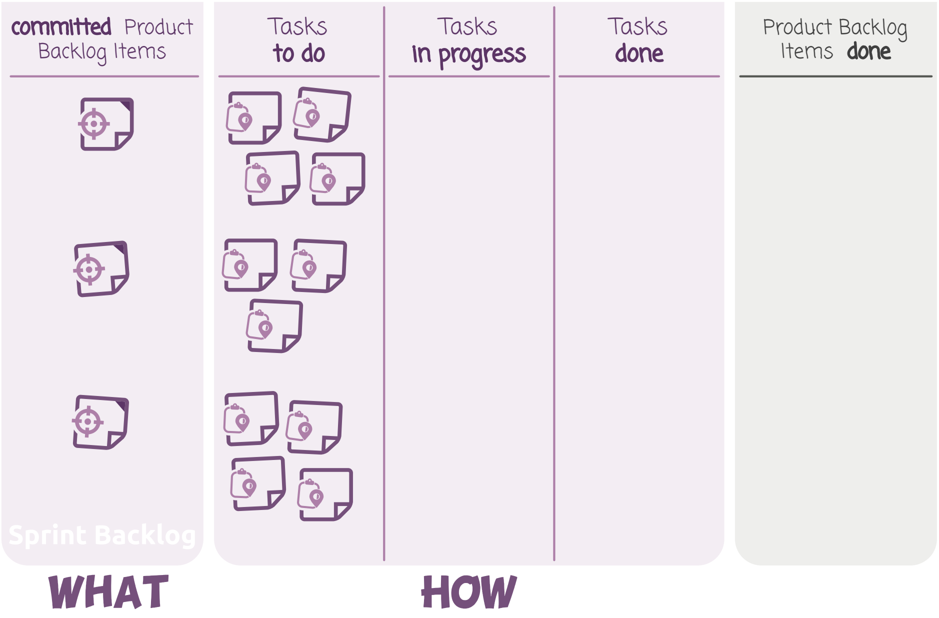
  1. Selecting and committing the team to a **set of product backlog items** to develop in the sprint → Sprint Backlog
  2. Coming up with the corresponding **tasks**

AGENDA

1. Sprint goal [PO]
2. Discussion [team+PO]
3. PBI selection [team]
4. Task identification [team]

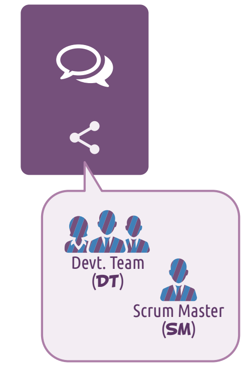
Artifacts

Sprint Board *after planning meeting*



committed Product Backlog Items	Tasks to do	Tasks in progress	Tasks done	Product Backlog Items done

Processes
Meetings



Daily Scrum (Meeting)

- Every day*, a.m. (15 minutes, stand up)
  * except on day 1
- Aims at reporting to each other:
  - **W**hat was done since the last Scrum meeting
  - **W**hat is intended to be done before the next Scrum meeting
  - **W**hat impedes team members, blocks their progress, reduces their effectiveness

AGENDA

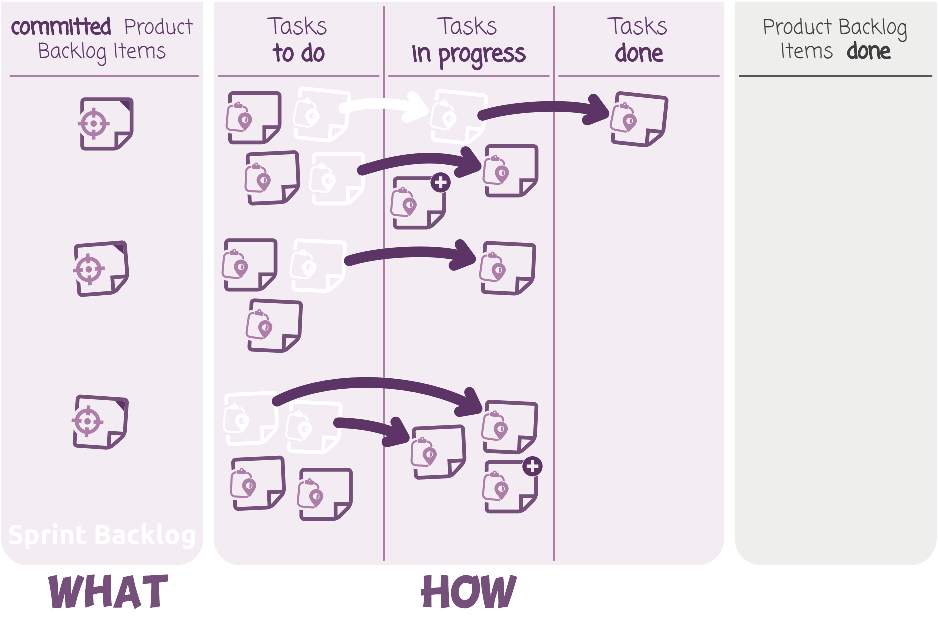
1. 3W-reports [team members]
   incl. changes onto the sprint board

Unique agenda item

a.k.a.
Daily Huddle
Daily Standup

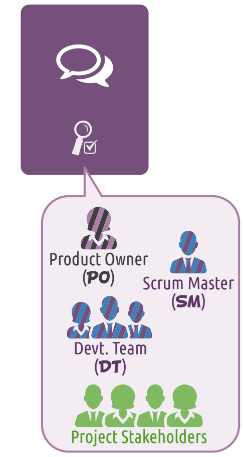
Artifacts

Sprint Board *during daily Scrum*



Processes

Meetings



Sprint Review Meeting

- On **day *n*** a.m. (2 hours, rather informal)
- Aims at:
  - Demonstrating the increment implemented in the sprint
  - Declaring which items are done
  - Getting feedback w.r.t. what was done

AGENDA

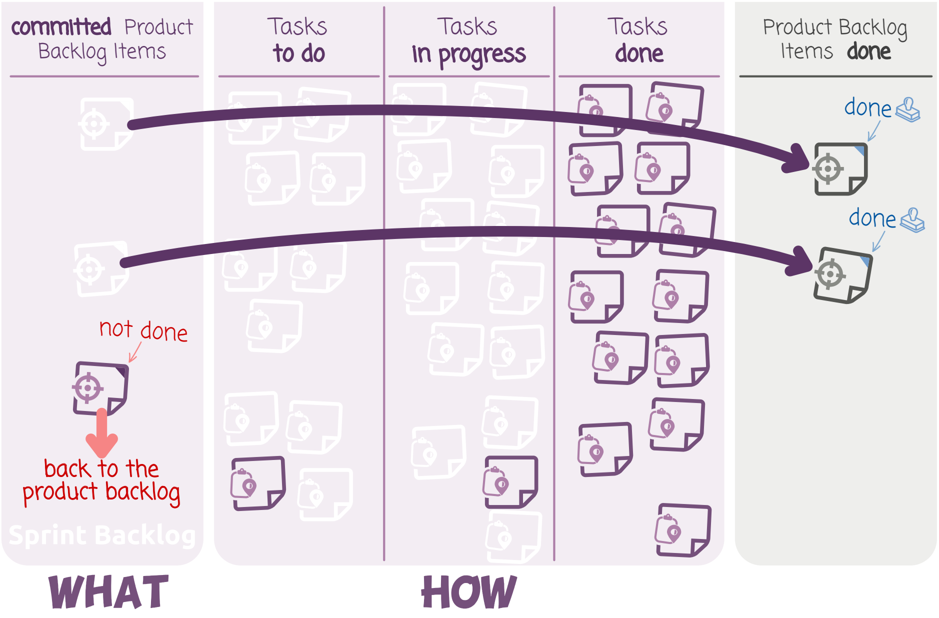
1. PSPI demonstration [team]
2. *Done-declaration* [PO]
3. Velocity tracking (optional)
4. Stakeholder feedback [all]

It is also possible that after the demo the PO realizes that the expected product shall be different. This should not be a source of frustration for the team!

a.k.a.
Sprint Demo Meeting

Artifacts

Sprint Board *during review meeting*


committed Product Backlog Items
Tasks to do
Tasks in progress
Tasks done
Product Backlog Items done
done
done
not done
back to the product backlog
Sprint Backlog
WHAT
HOW

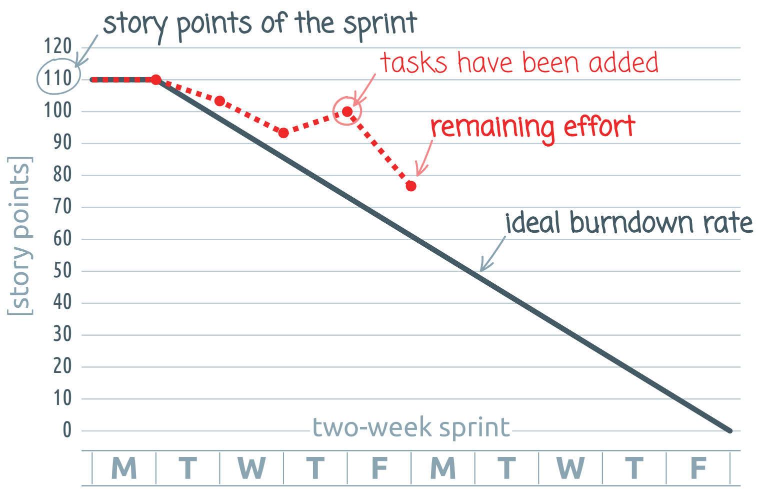
Definition of Done (DoD)

- Is a **shared understanding of expectations** that the increment (PSPI's) must live up in order to be releasable or deliverable*
- Is **specific to each team** and/or project
- Practically, is a clear and concise **list of broad requirements** the product/PSPI shall adhere to for the team to call it complete
- Is **consistent across all the items** of a product backlog
- Is somehow equivalent to the concept of **acceptance criteria**, but at a broader level
- Should be defined in the Project Management Plan

*Scrum.org/resources/scrum-glossary

Burndown Chart

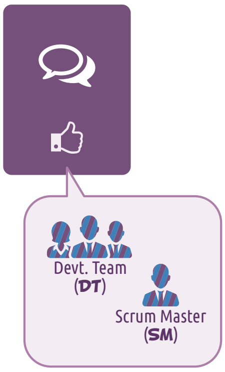
- Within a sprint, is a graphical representation of the work left to do versus time*



*en.wikipedia.org/wiki/burndown_chart

Processes

Meetings



Sprint Retrospective Meeting

- On **day *n*** p.m. (2 hours)
- Aims at collecting **feedback** on the **process**:
  - **W**hat went well
  - **W**hat could be improved
  - **W**hat did project participants learn
  - **W**hat still puzzles them

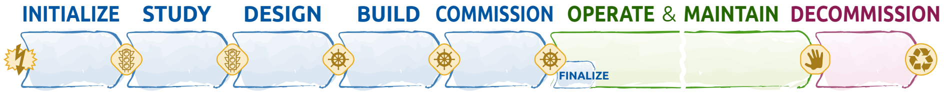
  and **deciding** what to do…
- By means of these retrospective meetings, the team shall **take ownership of their own process**

AGENDA

1. 4W-reports [team members]
2. Action plan [team members]

Embedding SCRUM in openSE

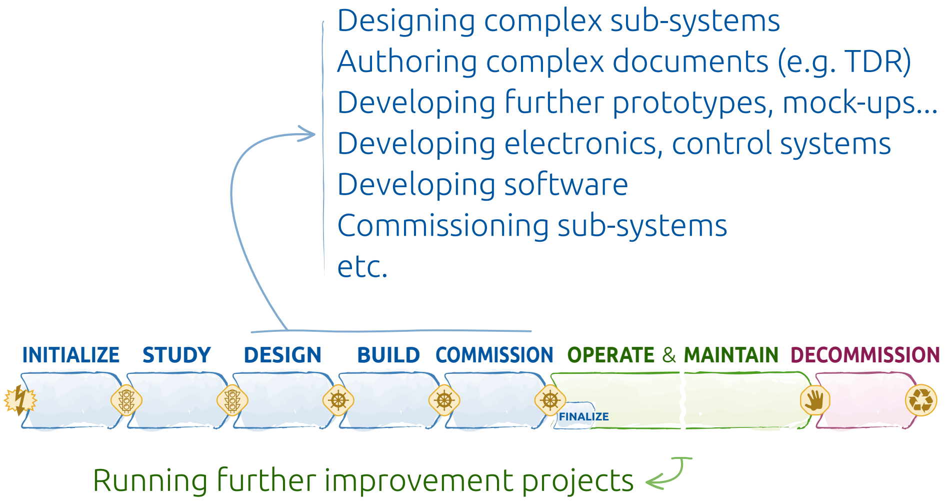
Authoring a Project Proposal/Roadmap



Gathering needs/user requirements
Working out solutions and concepts
Developing the product breakdown structure*
Defining product* requirements
Developing prototypes, mock-ups…
Authoring the Conceptual Design Report
etc.

* Product can be replaced by equipment, system, facility, process, service

Embedding SCRUM in openSE



Designing complex sub-systems
Authoring complex documents (e.g. TDR)
Developing further prototypes, mock-ups...
Developing electronics, control systems
Developing software
Commissioning sub-systems
etc.

Running further improvement projects

Lexique anglais - français

[en]	[fr]
Scrum	***Mêlée*** (Scrum)
Sprint	Sprint
Framework	Cadre de travail
Lifecycle	Cycle de vie
Roles	Rôles
Artifacts	Artéfacts
Processes	Processus
Events	Événements
Rules	Règles
Stakeholders	Parties prenantes
Product Owner	Responsable du produit
Development Team	Équipe de développement
Team member(s)	Équipier(s)
Scrum Master	*Maître de mêlée*
Product Backlog	Registre (de) produit
—— Item(s)	Élément(s) du ——
Sprint Backlog	Registre de/du sprint
Potentially shippable product incrément	Incrément de/du produit potentiellement livrable
Backlog Refinement Meeting	Réunion de définition du registre produit
Sprint Planning Meeting	Réunion de planification du sprint
Daily Scrum	*Mêlée quotidienne* (Scrum quotidien)
Sprint Review Meeting	Réunion de revue du sprint
Sprint Retrospective Meeting	Réunion rétrospective du sprint
Need/requirement	Besoin/exigence
User story	Récit utilisateur
Story card	Fiche de récit
Epic	Épopée
Story point(s)	Point(s) de récit
Task	Tâche
Velocity	Rapidité *(vélocité)*
Burndown Chart	Graphe d'avancement
Definition of Done	Définition du fait/fini
Acceptation Criteria	Critère d'acceptation

ℹ En pratique, les traductions en italique de sont pas utilisées.

References

The following resources were used to prepare this training session material :

Manifesto for Agile Software Development (2001) http ://AgileManifesto.org.

B. Boehm *et al.* (2003) *Balancing Agility and Discipline : A Guide for the Perplexed.* Boston, MA, USA : Addison-Wesley/Pearson Education. 368 p. ISBN 9780321186126

M. Cohn (2005) *Agile Estimating and Planning.* Upper Saddle River, NJ, USA : Prentice Hall. 368 p. ISBN 9780131479418

H. Kniberg (2007) *Scrum and XP from the Trenches. How We Do Scrum.* Toronto, Canada : InfoQ.com. 142 p. ISBN 9781430322641.

J. Patton, (2014) *User Story Mapping : Discover the Whole Story, Build the Right Product.* Sebastopol, CA, USA : O'Reilly. 324 p. ISBN 9781491904909

R. Pichler (2010) *Agile Product Management with Scrum : Creating Products that Customers Love.* Upper Saddle River, NJ, USA : Addison-Wesley Professional. 160 p. ISBN 9780321605788

K. S. Rubin (2012) *Essential Scrum : A Practical Guide to the Most Popular Agile Process.* Upper Saddle River, NJ, USA : Addison-Wesley Professional. 504 p. ISBN 9780137043293

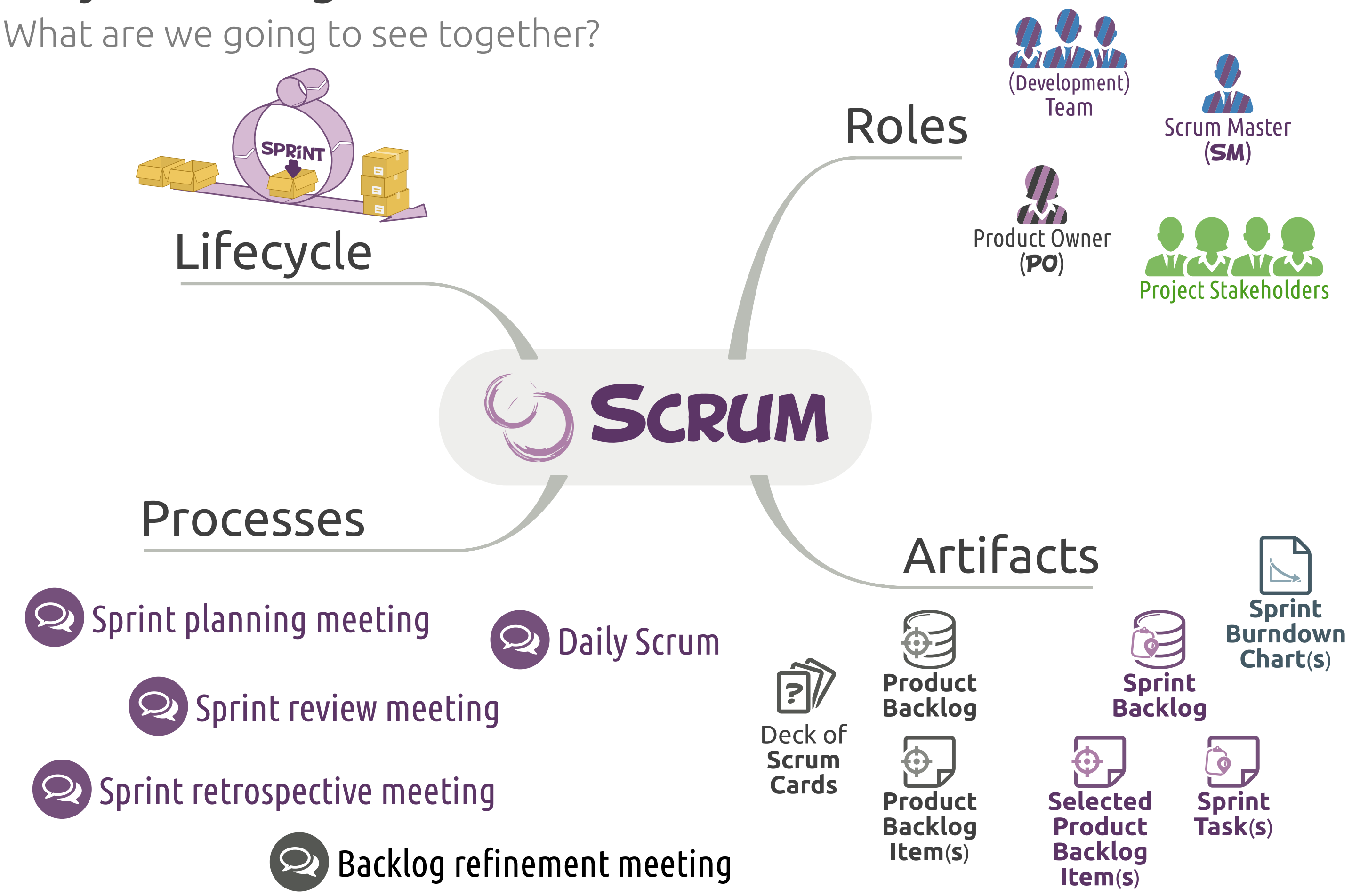
K. Schwaber, J. Sutherland (2017) *The Scrum Guide. The Definitive Guide to Scrum. The Rules of the Game.* Scrum.org and Scrum Inc. 17 p.

J. Sutherland (2014) *Scrum : The Art of Doing Twice the Work in Half the Time.* New York, USA : Crown Business. 256 p. ISBN 9780385346450.

The openSE editorial community (2014) *openSE Framework*, Geneva, Switzerland.

Project Management

What are we going to see together?


SPRiNT
Lifecycle
Roles
(Development) Team
Scrum Master (SM)
Product Owner (PO)
Project Stakeholders
SCRUM
Processes
Sprint planning meeting
Daily Scrum
Sprint review meeting
Sprint retrospective meeting
Backlog refinement meeting
Artifacts
Deck of Scrum Cards
Product Backlog
Product Backlog Item(s)
Sprint Backlog
Selected Product Backlog Item(s)
Sprint Burndown Chart(s)
Sprint Task(s)