

COPENHAGEN BUSINESS ACADEMY



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** 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.

five key values of the scrum framework

- Scrum Value #1. Courage
 - face difficult decisions that won't make everyone happy
- Scrum Value #2. Focus
 - the scrum team works on the right things in the correct order
- Scrum Value #3. Commitment
 - you should become **obsessed** with your goals.
 - Excuses should have no more room in your team
- Scrum Value #4. Respect
 - It's particularly crucial to properly receive and answer opposite opinions that the majority of the group do not agree with
- Scrum Value #5. Openness
 - there is no inappropriate opinion, decision, and action



How the customer explained it



How the project leader understood it



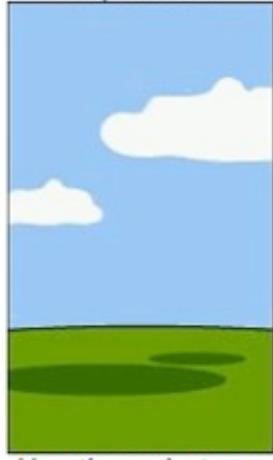
How the engineer designed it



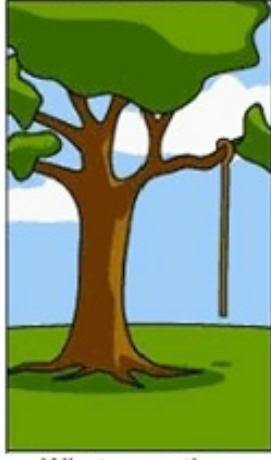
How the programmer wrote it



How the sales executive described it



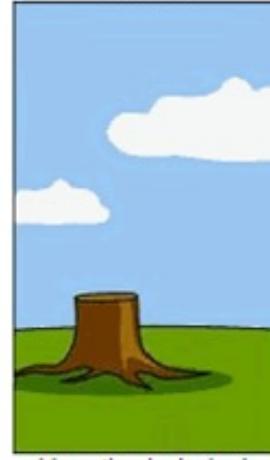
How the project was documented



What operations installed



How the customer was billed



How the helpdesk supported it



What the customer really needed

Waterfall approach

1. Conception

This phase involves identifying goals to be achieved after the problem is solved, estimating benefits in the new system over the current system, and identifying other areas that are affected by the solution. This phase also involves and developing the business case for the project.

2. Initiation

Involves a macro level study of the customer requirements.

3. Analysis

Involves carrying out detailed study of the customer requirements and arriving at the exact requirements of the proposed system. The phase involves freezing the requirements before the design phase begins.

4. Design

Involves translating the identified requirements into a logical structure, called design that can be implemented in a programming logic.

5. Construction

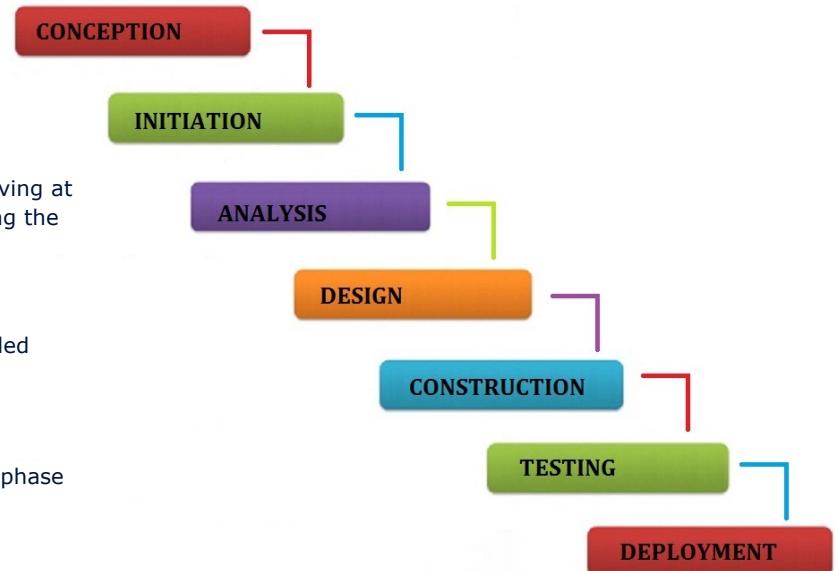
Involves integrating and testing all the modules developed in the previous phase as a complete system.

6. Testing and Integration

Involves integrating and testing all the modules developed in the previous phase as a complete system.

7. Implementation and maintenance

Involves converting the new system design into operation.

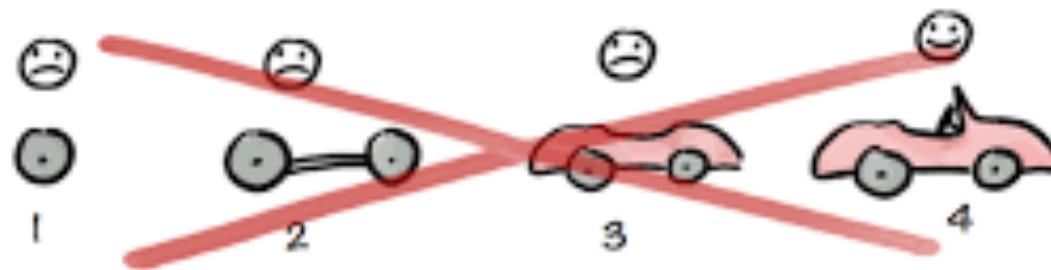


The waterfall approach assumes that requirements are stable and frozen across the project plan.

However, this is usually not true in case of large projects where requirements may evolve across the development process.

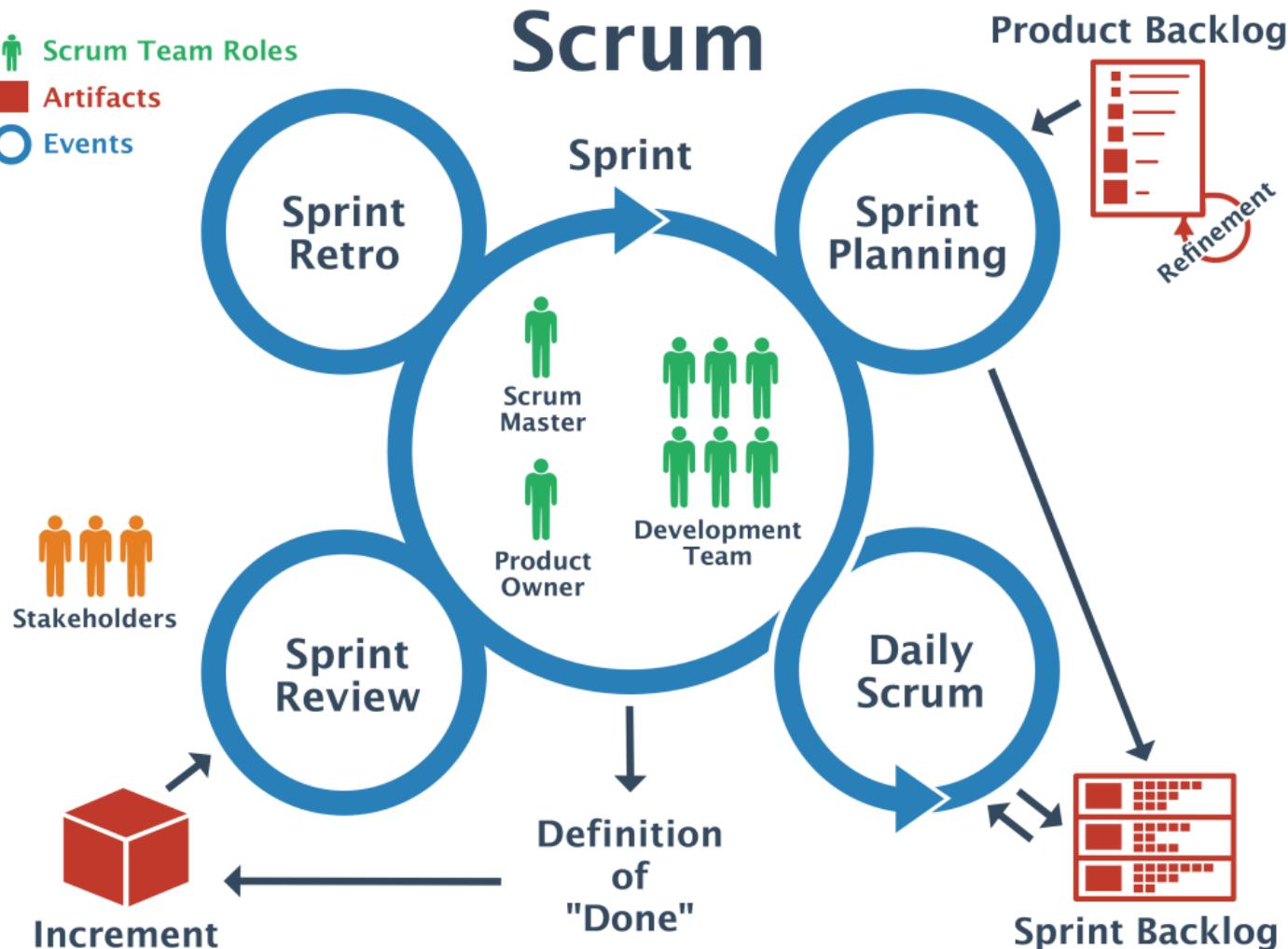
Spotify Development philosophy

Not like this....



Like this!





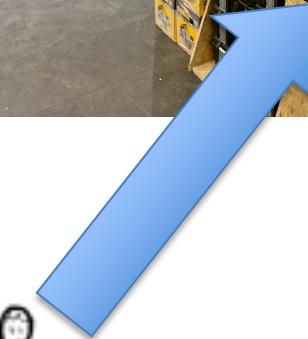
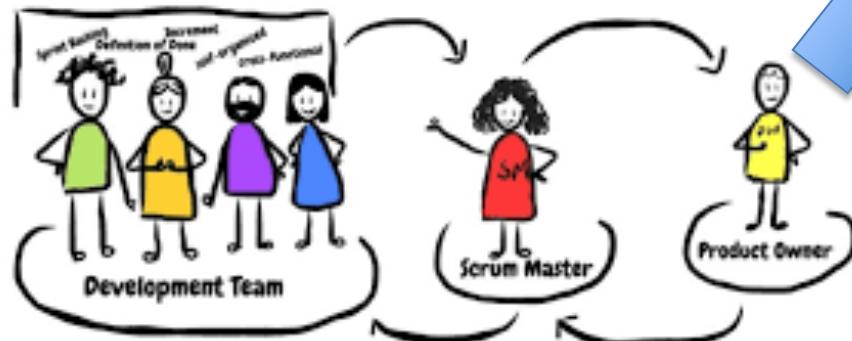
Vision: Et heatmap over kundernes bevægelser rundt i butikken



EPIC: En sensor som kan måle hvor populær en varegruppe er

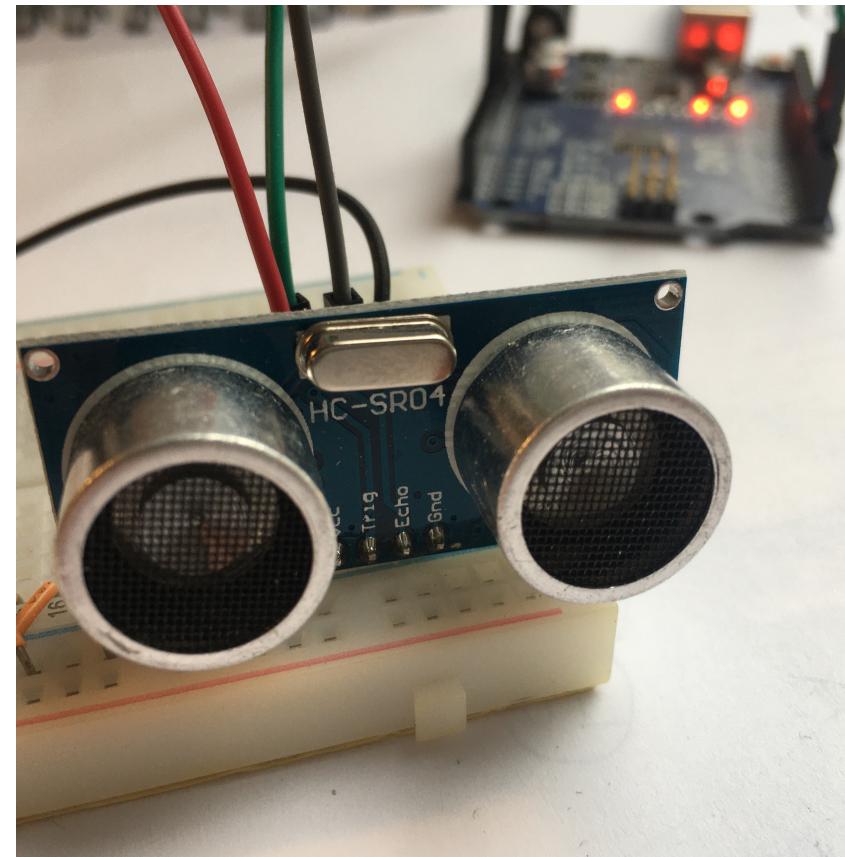


Scrum Team



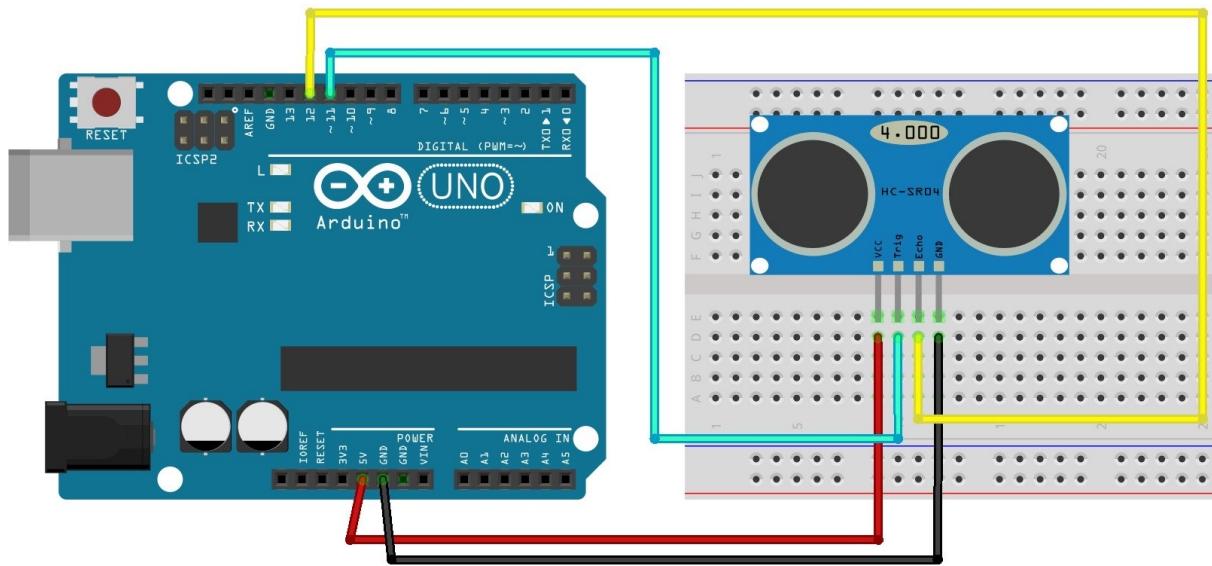
Fog bygning – Data-retrieval

- Design af IO-device
- Kodning af device
 - Arduino
 - R
- Analyse af data
- Design et dashboard



HC-SR04

- Installér Arduino
- Plug boardet
- Hent ino-kode fra github
- Hent R-kode fra github



Roles	Artifacts	Ceremonies
<ul style="list-style-type: none">• Product owner• Development team• Scrum master	<ul style="list-style-type: none">• Increment• Product backlog• Sprint backlog	<ul style="list-style-type: none">• Sprint planning• Sprint review• Sprint retrospective• Daily scrum

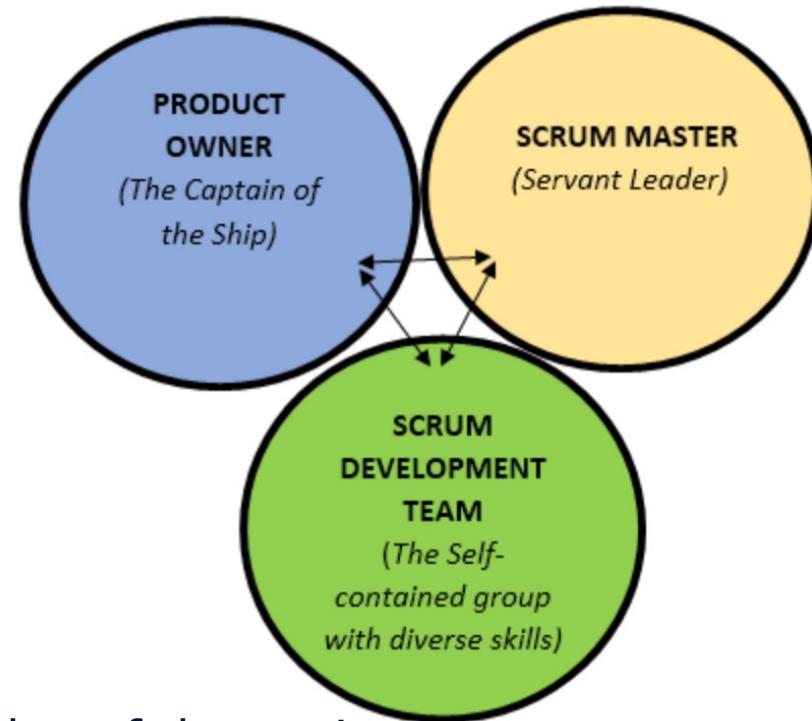
Roles

- Product owner
- Development team
- Scrum master

Many times we observed that the root cause of difficulties of a scrum team is either because these roles are not understood or they don't employ the right people.

Roles

- Product owner
- Development team
- Scrum master



Product Owner

Responsible for the business value of the project

Development Team

Self-organizes to get the work done

Scrum Master

Ensures that the team is functional and productive

Roles
<ul style="list-style-type: none">• Product owner• Development team• Scrum master

- Empowered and Autonomous,
- Cross-functional,
- Self-organized and small,
- Full-time participants,
- Working in the same room,
- One for all, all for one.

Development Team

Self-organizes to get the work done

- They have to breakdown the user stories, create tasks, define priorities and estimates, and they self-organize the implementation. In other words, they have to create, process, and deliver the Sprint Backlog.
- They have to perform Daily Scrum Meetings
- They have to ensure that at the end of the Sprint, potentially shippable product increment is delivered and demonstrated.
- They have to update the status and the remaining work efforts for their tasks

Roles
<ul style="list-style-type: none">• Product owner• Development team• Scrum master

Scrum Master

Ensures that the team is functional and productive

- To **coach** the Scrum Team to understand and **live the values** of the Scrum Framework
- To ensure efficient and close **collaboration** between the Scrum **Product Owner** and the Scrum Team
- To **remove impediments** which hinder the continuity of work
- To lead progress of work by **serving**
- To **moderate the Scrum Rituals** (Scrum Events).
- Enforces **time boxes**
- Keeps Scrum artifacts **visible**
- To **guard** the Scrum Team from external interference and interruptions while the team does work

Roles
<ul style="list-style-type: none">• Product owner• Development team• Scrum master

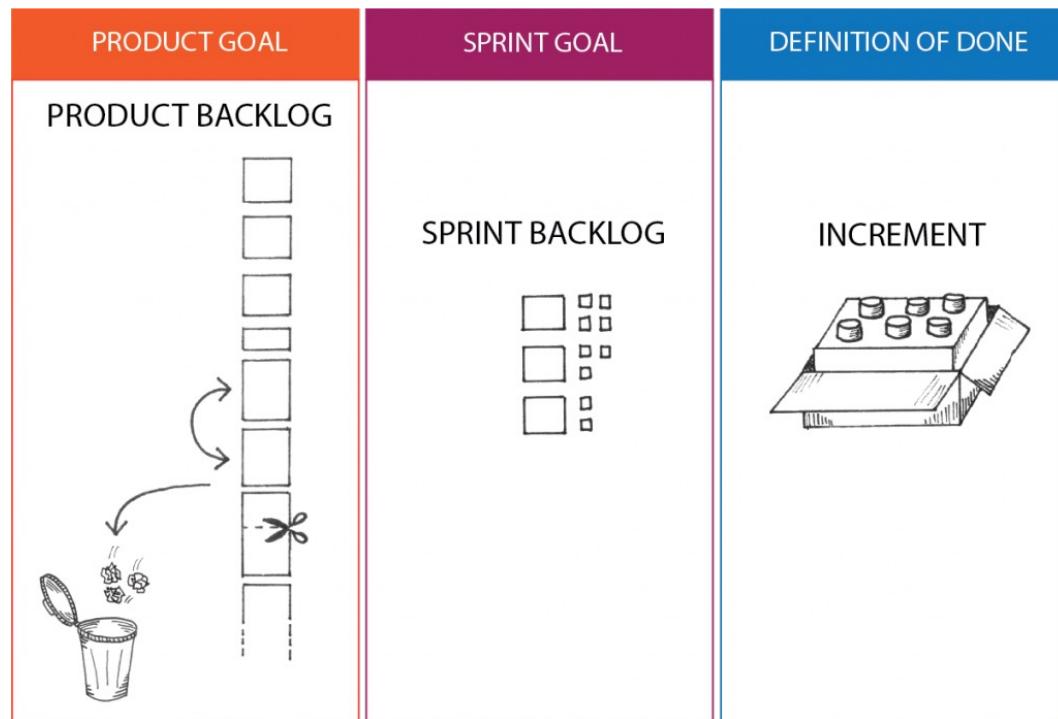
Product Owner

Responsible for the business value of the project

- **To manage and clarify project requirements**
- To guide releases and to ensure return on investment (ROI)
- To closely work with the Scrum Team and enable it to deliver the correct work on time
- **To manage stakeholders** and their expectations
- **To manage the Scrum Product Backlog.**

Artifacts

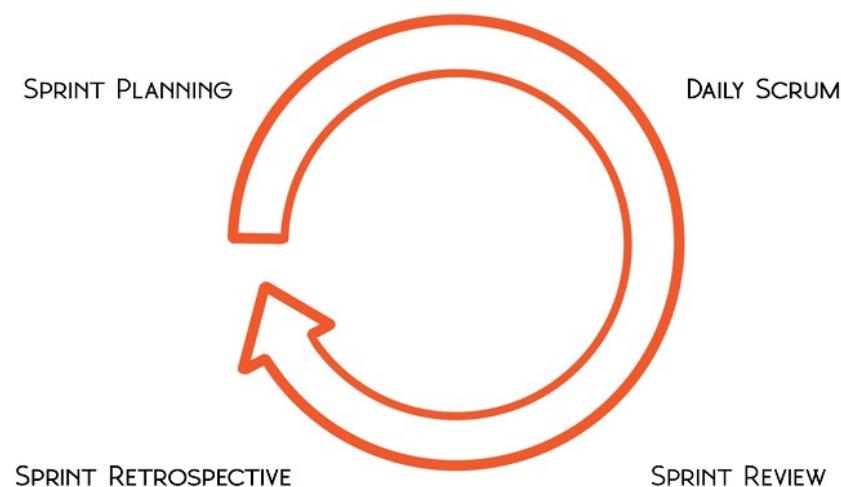
- Increment
- Product backlog
- Sprint backlog



Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum

THE SPRINT



*

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum

During the Sprint Planning Meeting, the Scrum Team **builds a viable Sprint Backlog**, which determines the **user stories** and **tasks** the team is going to implement until the end of this Sprint

WHAT-part:

The **Product Owner** introduces the **Sprint goal** and her **preselection** of requirements, which should go into the product increment.

The Scrum Team identifies the required **tasks** and their estimations. It's the role of the Scrum Team to decide which and how many requirements preselected by the product owner they can confidently deliver in this Sprint.

HOW-part:

fill **the Sprint Backlog** by identifying the concrete tasks needed to implement committed user stories for the Sprint. These tasks usually include activities such as **analysis, design, development** and **documentation**.

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum

The Scrum team demonstrates its work results.

The **Product Owner** controls whether the Scrum team delivered the requirements they had committed during the Sprint Planning Meeting accurately or not.

In Sprint Review Meetings, **everyone** is allowed to deliver their feedback about the demonstrated product increment.

Participants of the Sprint Review Meeting are the Scrum Team, the Scrum Product Owner, the Scrum Master. Optionally all other stakeholders

Ceremonies

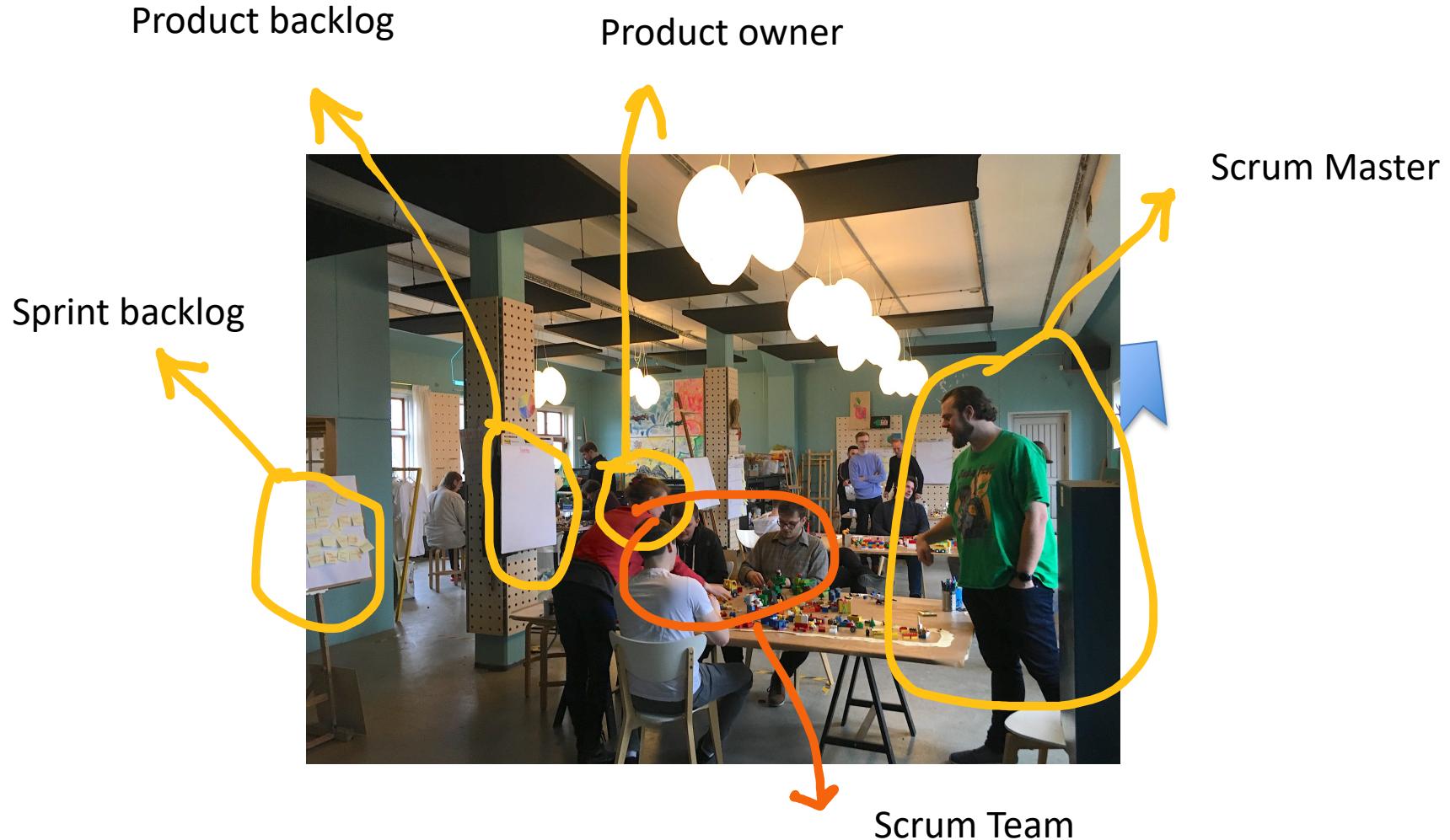
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum

The Sprint Retrospective Meeting happens directly after the Sprint Review Meeting, and it closes the Sprint.

Without this meeting, the Scrum Team will **never be able to improve** its overall throughput, and they cannot focus on the improvement of team performance

Without exception, the Scrum team conducts Sprint Retrospective Meetings at the end of every Sprint

EPIC: En svævebane med plads til 20 ned over opalsøen



Product Backlog Item

Often called (user) story, or just PBI.

Examples of user stories

- As a **securityofficer**, I want to **have a complete checklist** so that **we can avoid accidents**.
- As a **guest**, I want to **have an open view of the terrain** so that **I can take nice pictures**.
- As a **guest**, I want to **have a smoothe ride** so that **I won't get wet**.
- As an **operator**, I want to **have a way to stop the cart** so that **we can avoid accidents**.
- As a **manager**, I want to **have a testrun** so that **I can convince my stakeholders**.

Product Backlog - Living document

The Scrum Product Backlog is changed throughout the whole project.

If needed, new requirements are **added** and existing requirements may be **modified**, defined in more detail or even **deleted**.

Requirements are no longer frozen early on. Instead the final set of requirements within the Scrum Product Backlog is also developed iteratively, together with the resulting software.

This but allows **is different to traditional requirements engineering** s maximizing customer value and minimizes development effort.

When using the Scrum Framework about **10% of the Scrum Teams total time** should be reserved for maintaining the Scrum Product Backlog (discussion, estimation etc.).

Scrum Master

- Facilitates the Scrum process
- Helps resolve impediments
- Creates an environment conducive to team self-organization
- Captures empirical data to adjust forecasts
- Shields the team from external interference and distractions to keep it in group flow
- Enforces time boxes
- Keeps Scrum artifacts visible
- Promotes improved engineering practices
- Has no management authority over the team (anyone with authority over the team is by definition not its Scrum Master)



**Scrum
Master**

Scrum Development Team

- Cross-functional (e.g., includes members with testing skills, and often others not traditionally called developers: business analysts, domain experts, etc.)
- Self-organizing / self-managing, without externally assigned roles
- Negotiates commitments with the Product Owner, one Sprint at a time
- Has autonomy regarding how to reach commitments
- Intensely collaborative
- Most successful when located in one team room, particularly for the first few Sprints
- Most successful with long-term, full-time membership. Scrum moves work to a flexible learning team and avoids moving people or splitting them between teams.
- 7 ± 2 members
- Has a leadership role

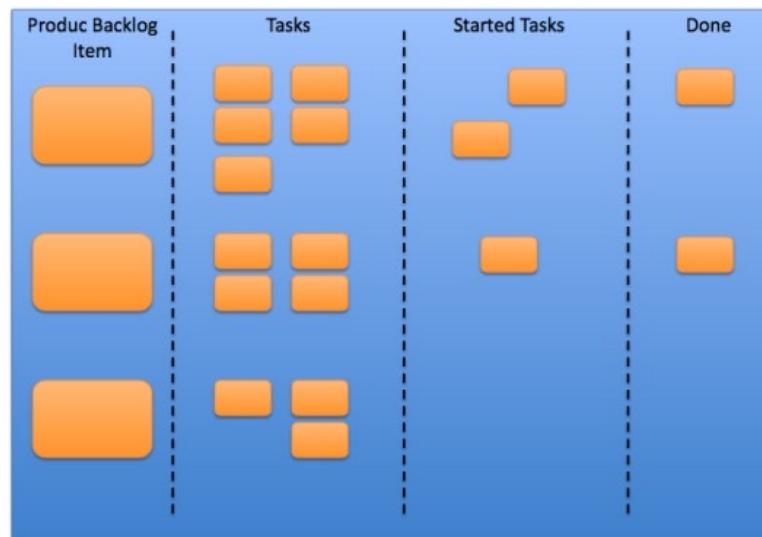


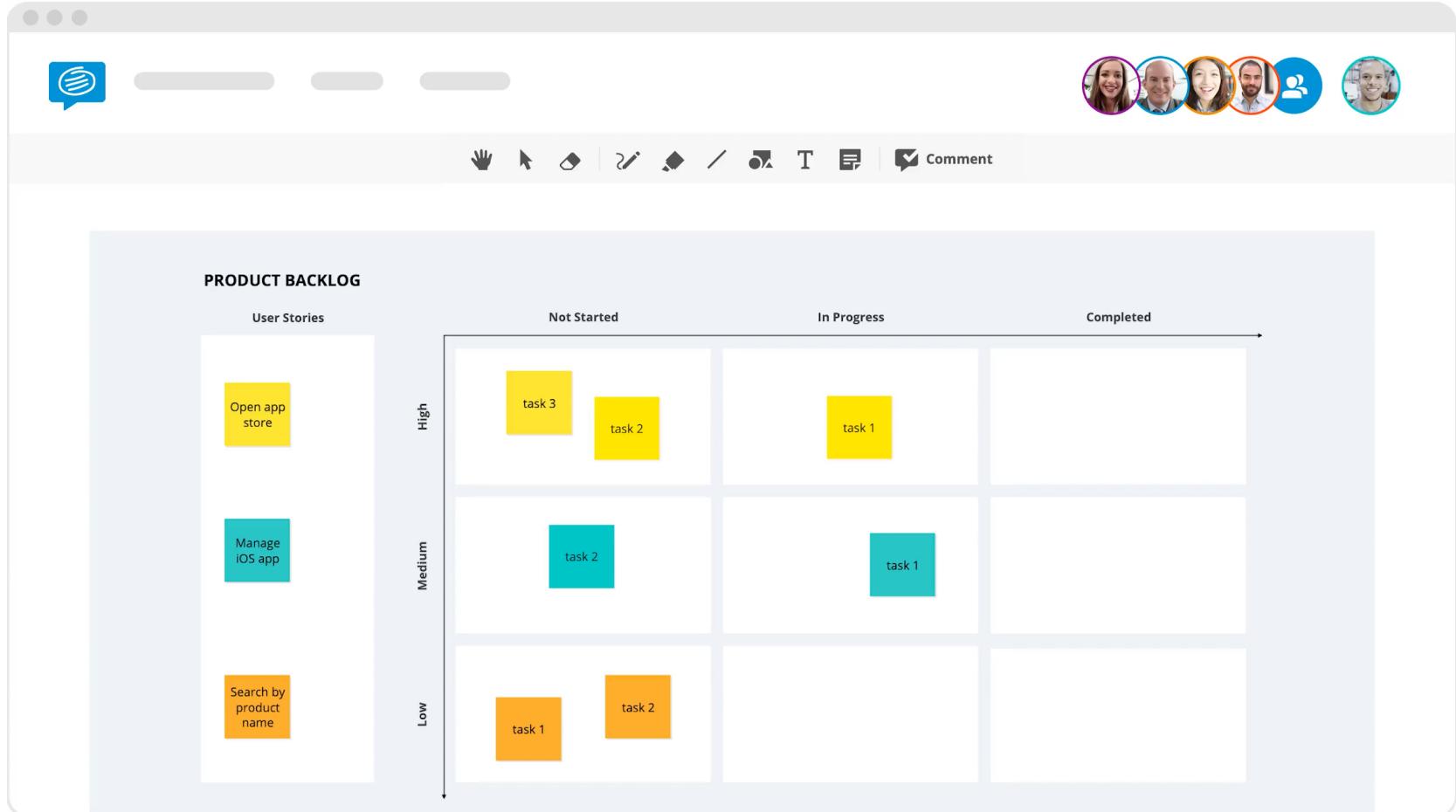
The Team



Sprint Backlog

- Contains **committed stories** negotiated between the team and the Product Owner during the Sprint Planning Meeting
- **Initial tasks** are identified by the team during Sprint Planning Meeting
- Team will discover **additional tasks** needed to meet the fixed scope commitment during Sprint execution





The screenshot shows a digital interface for managing a product backlog. On the left, a sidebar lists "User Stories" with three items: "Open app store" (yellow), "Manage iOS app" (teal), and "Search by product name" (orange). To the right is a matrix grid for tracking tasks. The vertical axis represents priority levels: High, Medium, and Low. The horizontal axis represents status: Not Started, In Progress, and Completed. The grid contains the following tasks:

User Story	Priority	Status	Task ID
User Stories	High	Not Started	task 3
		Not Started	task 2
		In Progress	task 1
User Stories	Medium	Not Started	task 2
	Medium	In Progress	task 1
User Stories	Low	Not Started	task 1
	Low	Not Started	task 2

The interface includes a toolbar at the top with various icons (hand, arrow, etc.) and a "Comment" button. A group of five user avatars is visible in the top right corner.

User Story

- As a **<role>** I can **<activity>** so that **<business value>**
- ... is short, simple description of a feature told from the perspective of the person who desires the new capability (typically user or customer)
- formal template

As a registered user,

I want to reset my password,

so that I can get back into the site if I forget my password

As **who**,
I want **what**,
so that **why**.

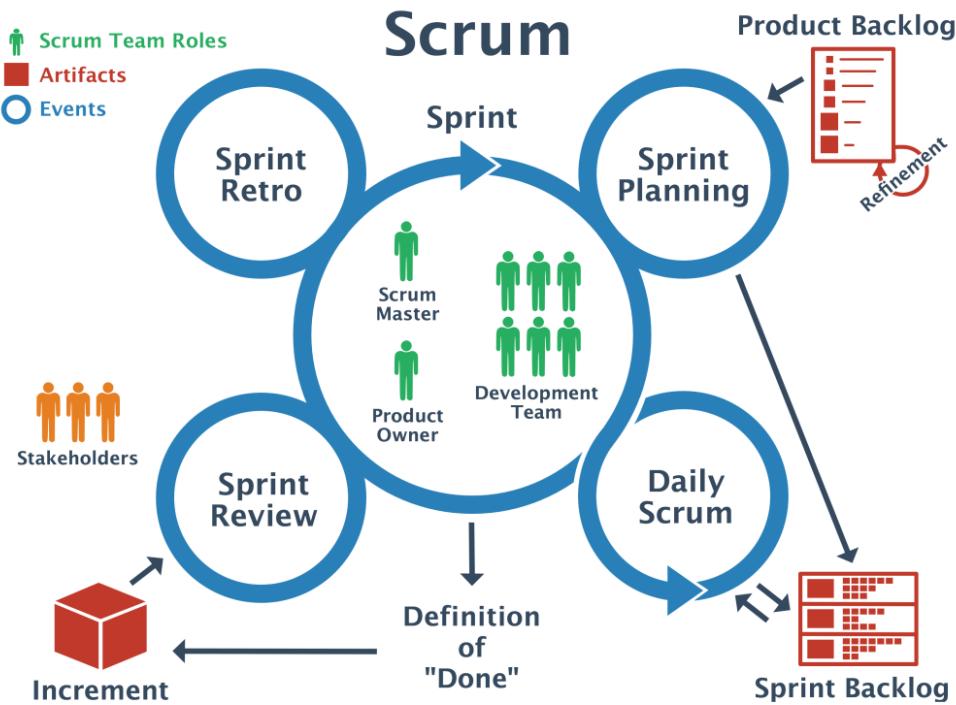
Writing good User Stories

I
N
V
E
S
T

ndependent
egotiable
alueable
stimable
mall / appropriately sized
estable

- Rollerne (SM, PO, DT)
 - Skal de ændres til i morgen?
- Artefakterne (PBL,SBL, Increment)
 - Userstory->task, konkrete nok? Ejerskab?
- Events (Planning, doing, review, retro?)
 - Fik man det gjort?

EVALUERING



Roles	Artifacts	Ceremonies
<ul style="list-style-type: none">• Product owner• Development team• Scrum master	<ul style="list-style-type: none">• Increment• Product backlog• Sprint backlog	<ul style="list-style-type: none">• Sprint planning• Sprint review• Sprint retrospective• Daily scrum

