Scrum

León Jaramillo

softserve

Software Development Lifecycle

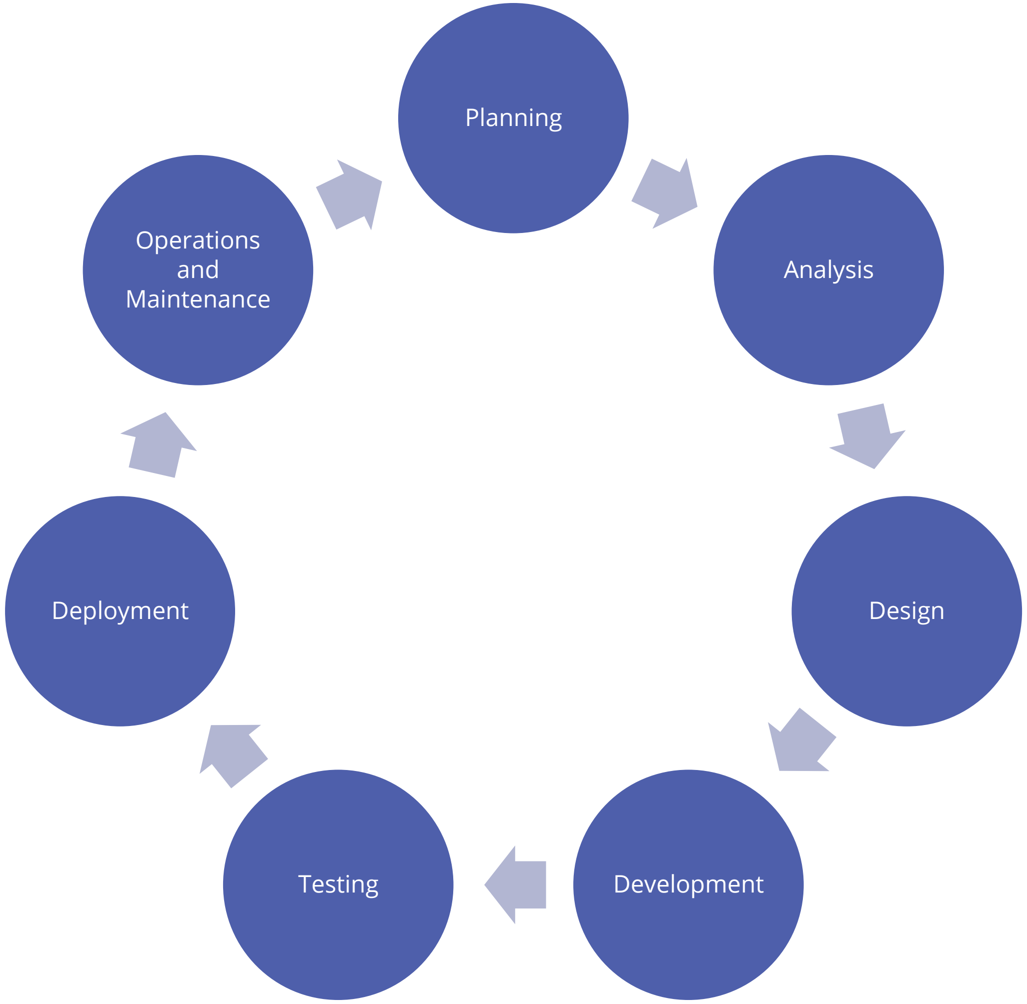
Scrum

Software Development Lifecycle (SDLC)

- The **Software Development Lifecycle (SDLC)** is a process used for developing software systems.

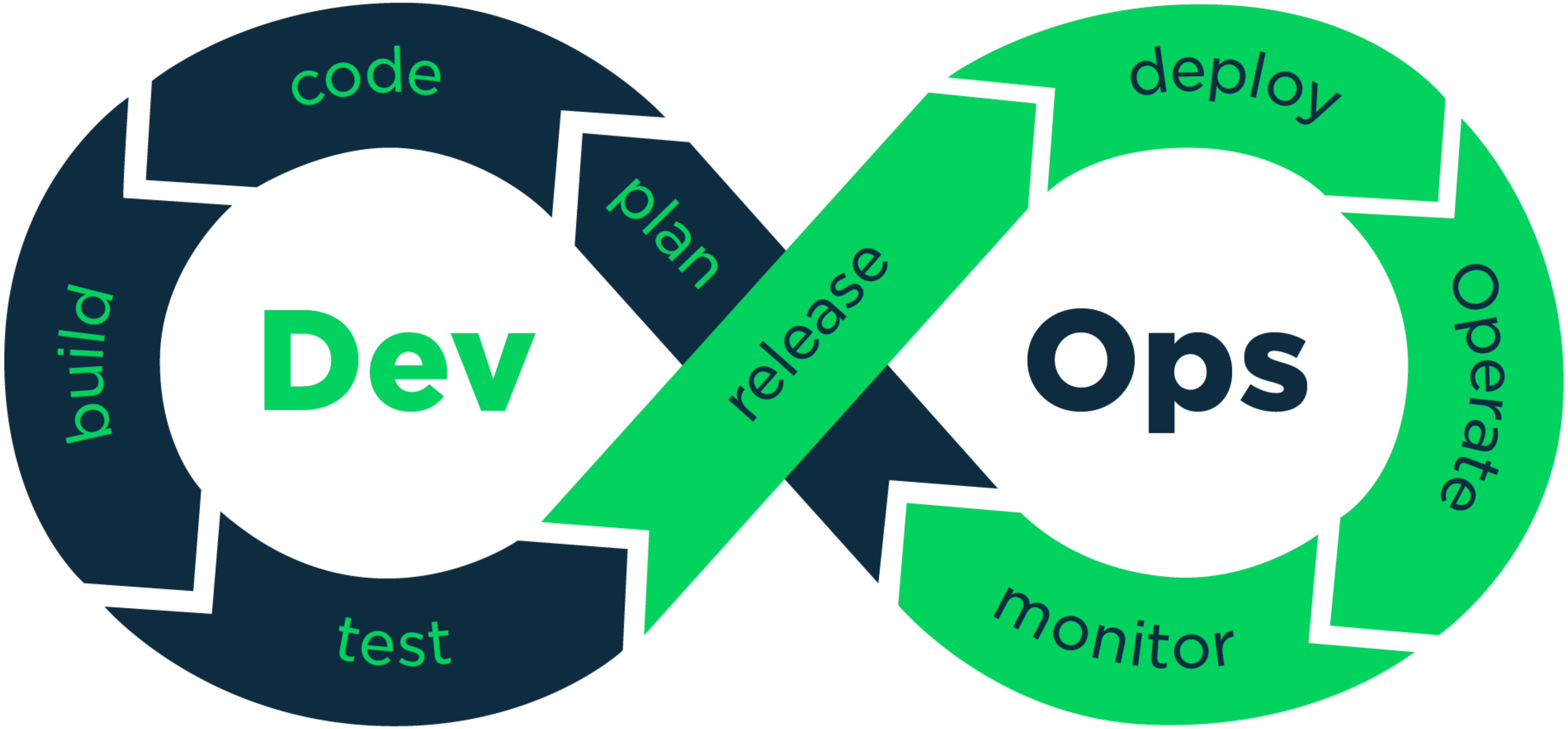
- Its phases comprise the **software development process**.

- The for a software system, the SDLC follows a defined **model**, implemented using a **methodology**.



softserve

SDLC in DevOps



https://danielmaldonado.com.ar/programacion/que-es-devops/

softserve

Project Management

Scrum

softserve

Project Management

- **Project Management** involves leading the work of a team to achieve all the goals of a project within de given constraints.

- The main constraints usually are **time**, **budget** and **scope**.

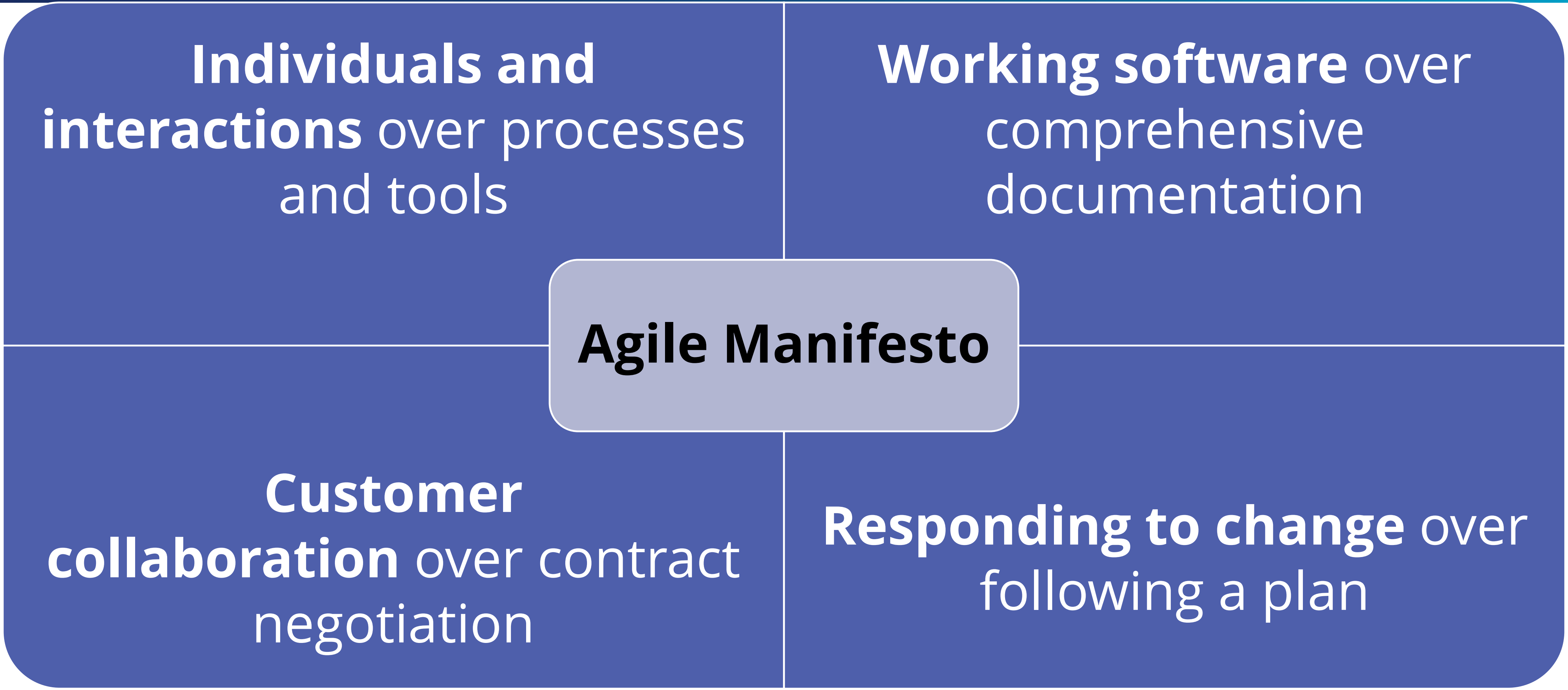
- A **project** is a temporary and unique endeavor designed to produce a product, service, or result with a defined beginning and end.

- Projects and project management are **present** in many different disciplines. Notably, they're present in engineering.

Project Management in Software Development

- Software development features different project management **methodologies and frameworks**, so we can build a software product in a systematic way.
- There is **no methodology or framework** that is suitable for every kind of software product, although there is a consensus on Agile's good outcomes.
- We could name several methodologies/frameworks:
  - Waterfall
  - Spiral (RUP and others)
  - V-Model
  - eXtreme Programming (XP)
  - **Agile** (**Scrum**, Kanban, among others)
- We've gone from a **predictive** to an **empirical approach**.

Agile



Individuals and interactions over processes and tools

Working software over comprehensive documentation

Agile Manifesto

Customer collaboration over contract negotiation

Responding to change over following a plan

Kanban (briefly)

Kanban

Kanban is a Lean workflow management method that helps organizations to improve and manage work systems.

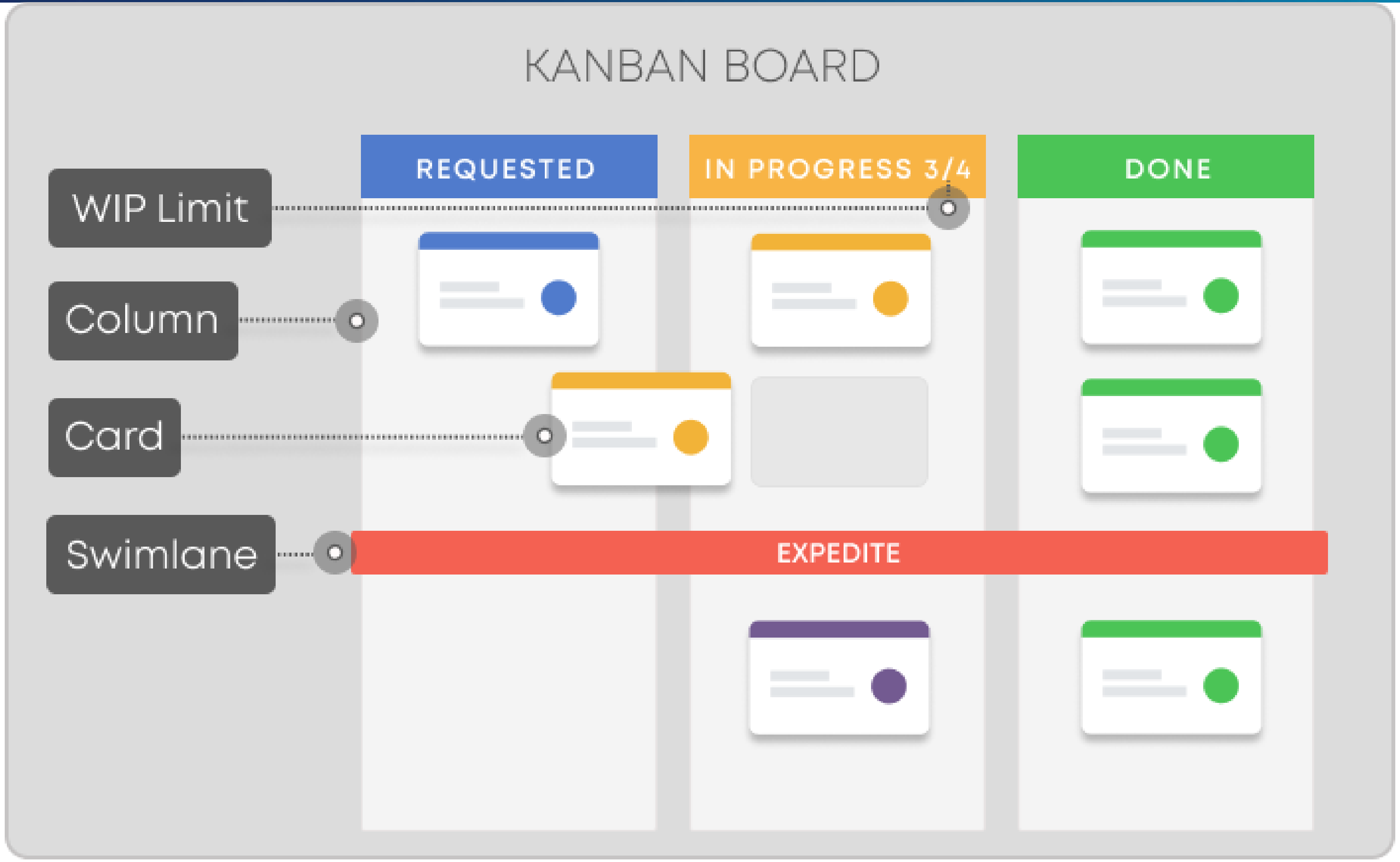
It is also a **framework** used to implement agile DevOps software development. Fits naturally with **Continuous Delivery**, since both comprise **just-in-time delivery**.

It was born as a **Lean** practice within the **Toyota Production System (TPS)**.

Kanban relies on a set of **practices**:

- Visualizing the workflow (via a **Kanban board**)
- Limiting work in progress (WIP)
- Managing flow
- Making process policies explicit
- Implementing feedback loops
- Improving colaboratively

Kanban Board



From: kanbanize.com

softserve

Scrum

Do you use Scrum?

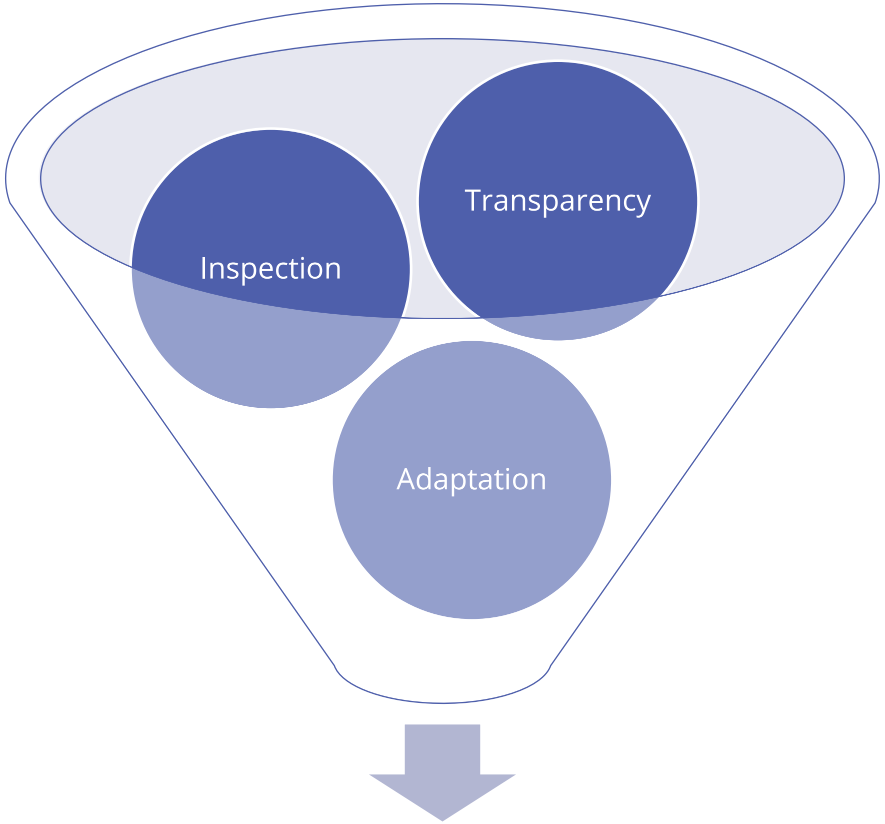
What is Scrum?

- **Scrum** is a framework for developing and sustaining **complex products**.

- It is also a framework that **helps teams** to work together.

- It has had a **long history** (despite its relatively recent popularity):
  - Hirotaka Takeuchi and Ikujiro Nonaka introduced the term to software development in 1986
  - Ken Schwaber and Jeff Sutherland defined the framework in their 1995 paper
  - Schwaber and Sutherland published the Scrum Guide since 2009
  - Now it‘s widespread in different industries

Which are the Scrum pillars?



Scrum Theory

softserve

What are the Scrum values?

Commitment to achievement and support

Focus on the work of the sprint

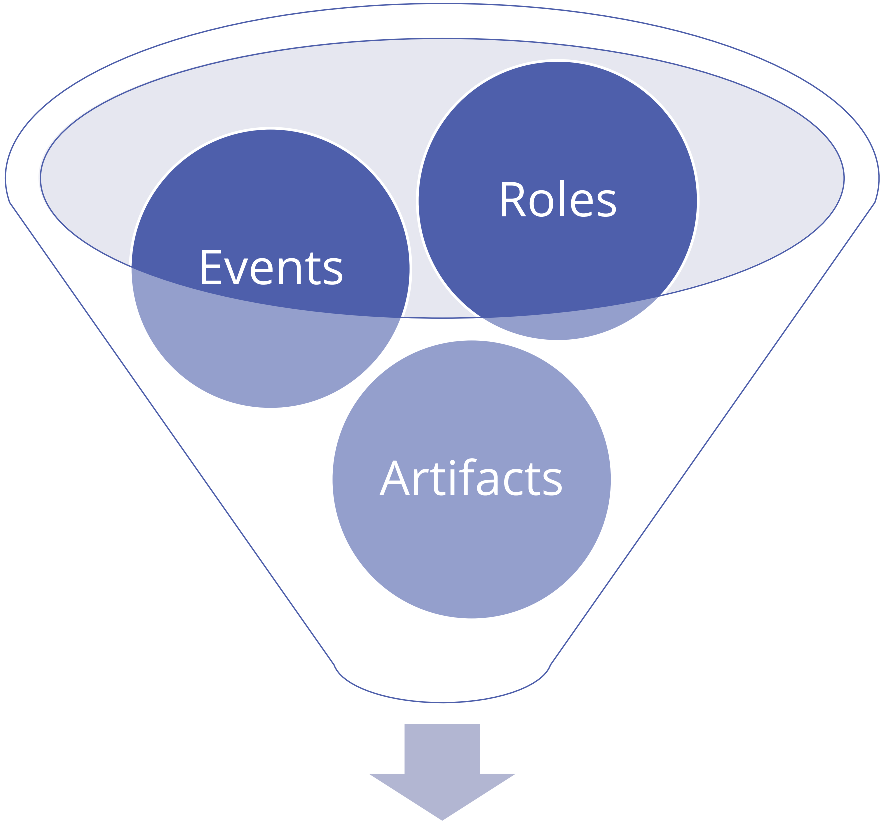
Openness about the work and the challenges

Respect within the team

Courage to do the right thing and work on tough problems

softserve

Which are the Scrum key elements?

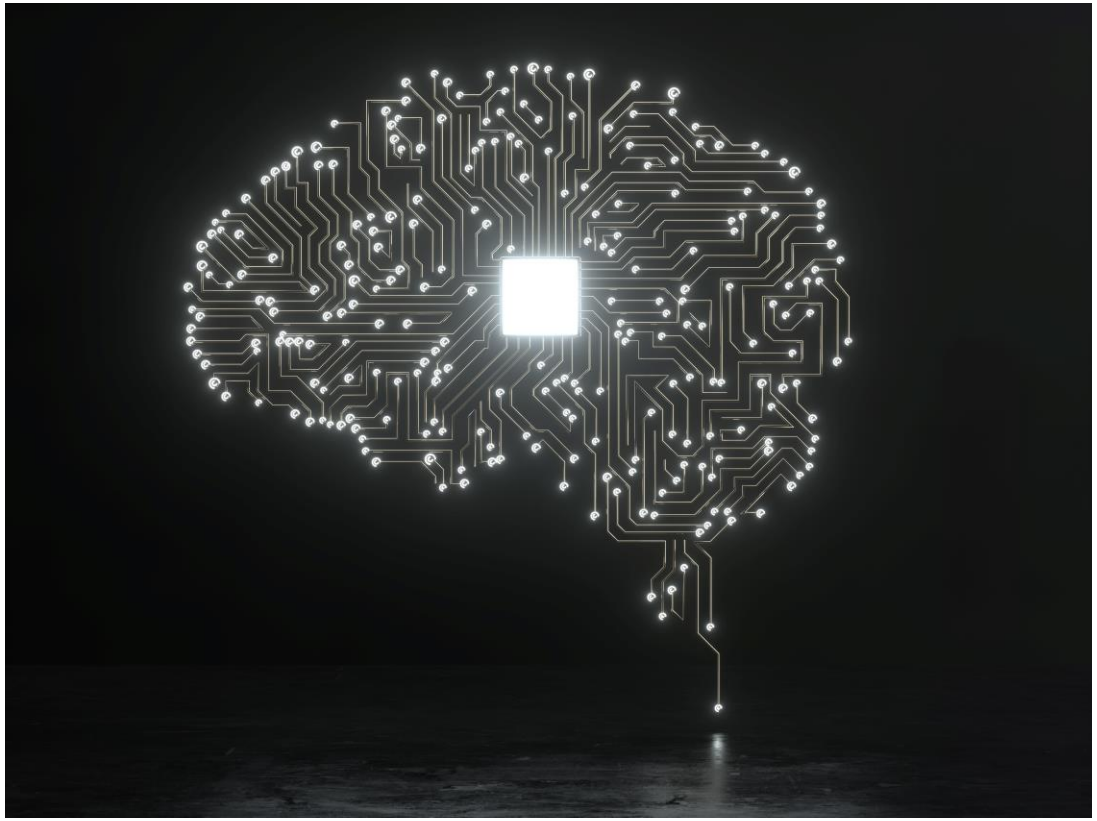


Scrum Key Elements

Scrum Roles

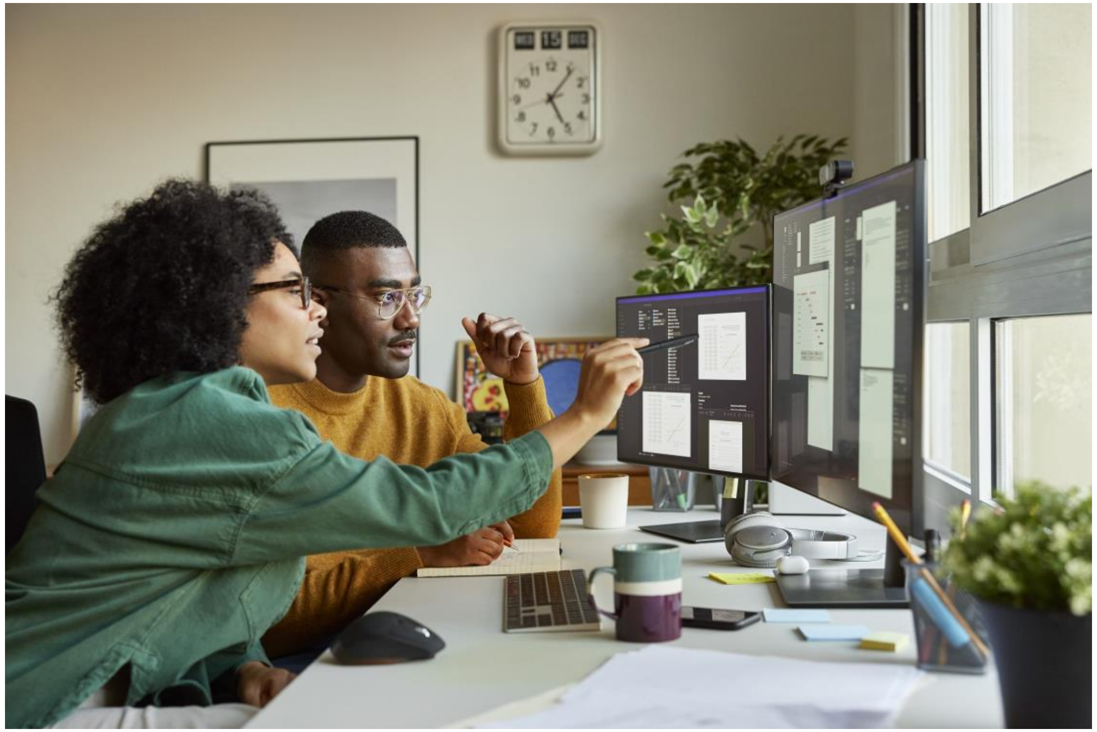
Developers

- **Developers** are the members of the Scrum team that create the usable increments each sprint.
- They should create the **Sprint Backlog**, adhere to **DoD** and **work** daily **towards** the sprint goal.



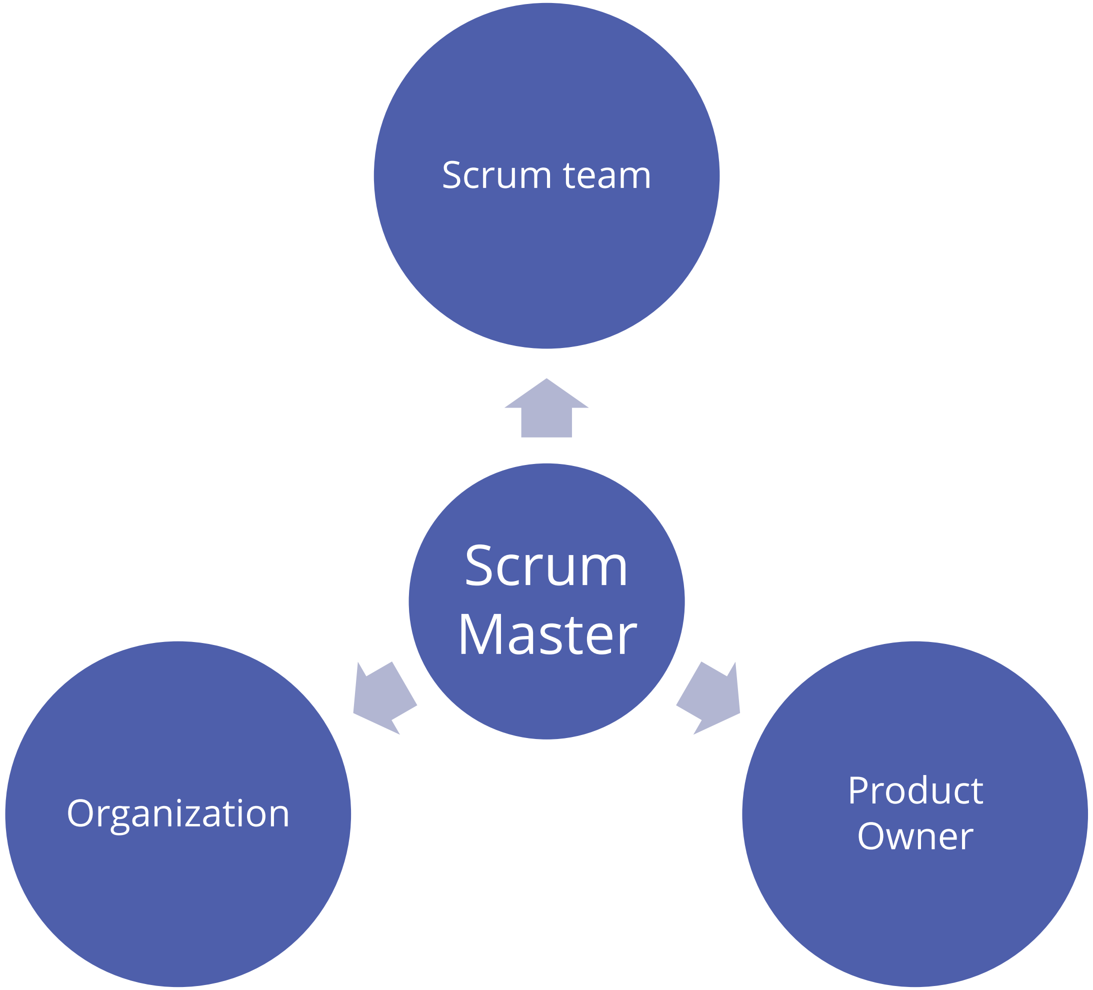
Product Owner (PO)

- The **Product Owner** (PO) should **maximize the value** delivered by the Scrum team.
- He **manages** the Product Backlog.
- He communicates the **Product Goal**.
- He represents the interests of many **stakeholders** in the Product Backlog.



Scrum Master

- The **Scrum Master** is accountable for the Scrum team's effectiveness using Scrum.
- He works on **improving** the team's practices.
- The Scrum Master serves to the team, the Product Owner and the whole organization while adopting and using Scrum.

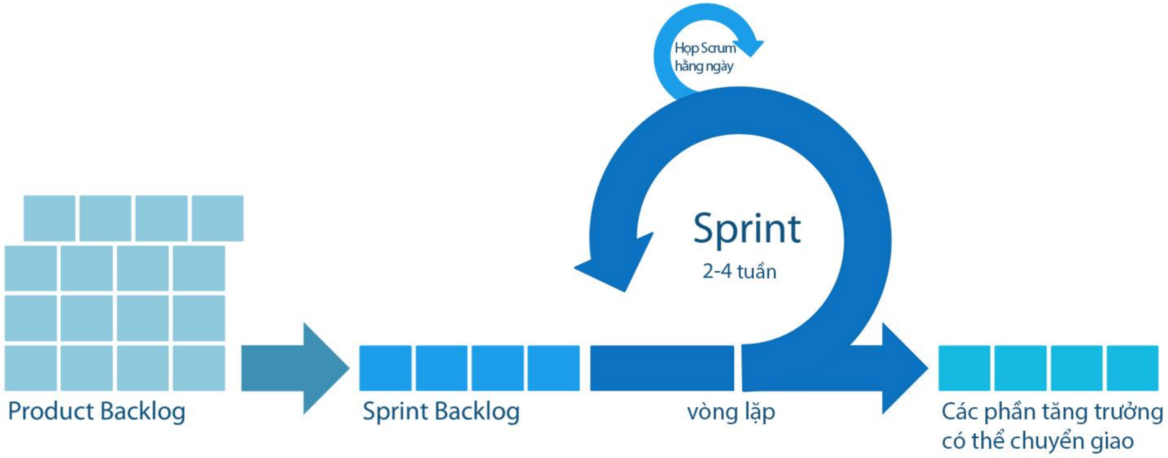


softserve

Scrum Events

Sprint

- A **sprint** is a fixed length (1 month or less) Scrum event.
- All the **work needed** to reach the Product Goal is done within subsequent sprints.
- Its defined time-box enables both: **predictability** and **adaptability**.



softserve

Sprint Planning

- Sprint Planning **initiates the sprint** by laying out the work to be performed for the sprint.
- Sprint Planning addresses the following **topics**:
  1. Why is this Sprint valuable?
  2. What can be Done this Sprint?
  3. How will the chosen work get done?
- The **Sprint Goal** and the **Sprint Backlog** should be clear after this meeting.
- Here are useful some **estimation techniques**, such as *Planning Poker*, *T-Shirt Sizes* or the *Bucket System*.

Daily Scrum (or Daily Meeting)

- The **Daily Scrum** is a 15-minute event for the Developers of the Scrum Team.
- The **purpose** of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary.
- Here we may **track** work using chart like Burn-down charts.

What have you completed since the last meeting?

What will you do until the next meeting?

Is there anything blocking your progress?

Sprint Review

- The **Sprint Review** is the second to last meeting in the sprint.
- The **purpose** of the Sprint Review is to inspect the **outcome** of the sprint and determine future adaptations.
- It is **not** just a presentation.
- Stakeholders from outside of the Scrum team often **attend** this event.

Sprint Retrospective

- **Sprint Retrospective** is the meetings that finishes the sprint and is done by the Scrum team.
- Its **purpose** is to **plan** ways to increase quality and effectiveness.
- It is desirable that stakeholders from outside of the Scrum team **don’t attend** this event.

Event Time Boxes

- **Event time boxes** are the maximum amounts of time that a Scrum team takes to execute Scrum events.
- They are **agreed** in advance. Possibly considering **recommended** time boxes.

Event / Sprint	30 days	3 weeks	2 weeks	1 week
Sprint planning	8 hours	< 8 hours	< 8 hours	< 4 hours
Daily scrum	15 minutes			
Sprint review	4 hours	< 4 hours	< 4 hours	< 2 hours
Sprint retrospective	3 hours	< 3 hours	< 3 hours	< 1.5 hours

Scrum Artifacts

Product Backlog

- The **Product Backlog** is an emergent, ordered list of what is needed to improve the product.

- It is the **single source of work** undertaken by the Scrum team.

- **Product Backlog refinement** is the act of breaking down and further defining Product Backlog items into smaller more precise items.

- Its commitment is the **Product Goal**.

- The **Product Owner manages** this artifact.

Sprint Backlog

- The **Sprint Backlog** is composed of
  - the **Sprint Goal** (why),
  - the set of **Product Backlog items selected** for the Sprint (what),
  - as well as an **actionable plan** for delivering the Increment (how).

- Its commitment is the **Sprint Goal**.

- The **Developers manage** this artifact.

- The Sprint Backlog can be updated **throughout the sprint**, but the Sprint Goal should remain always clear.

Product Backlog Items

User stories

Use cases

Feature definitions

Non-functional requirements

Constraints

Bugs

User Stories

- A **user story** is an informal, general explanation of a software feature written from the perspective of the end user.
- User stories are **building blocks** of larger agile items like epics and initiatives.
- Although it is a way to specify an user **requirement**, it is **more** than that.

As a [role]
I want [goal]
So that [benefit]

Acceptance criteria:

…
…
…
…

Increment

- An **Increment** is a concrete steppingstone toward the Product Goal.
- Each Increment is **additive** to all prior Increments.
- In order to provide value, the Increment must be **usable**.
- Its commitment is the **Definition of Done**.

Definition of Done

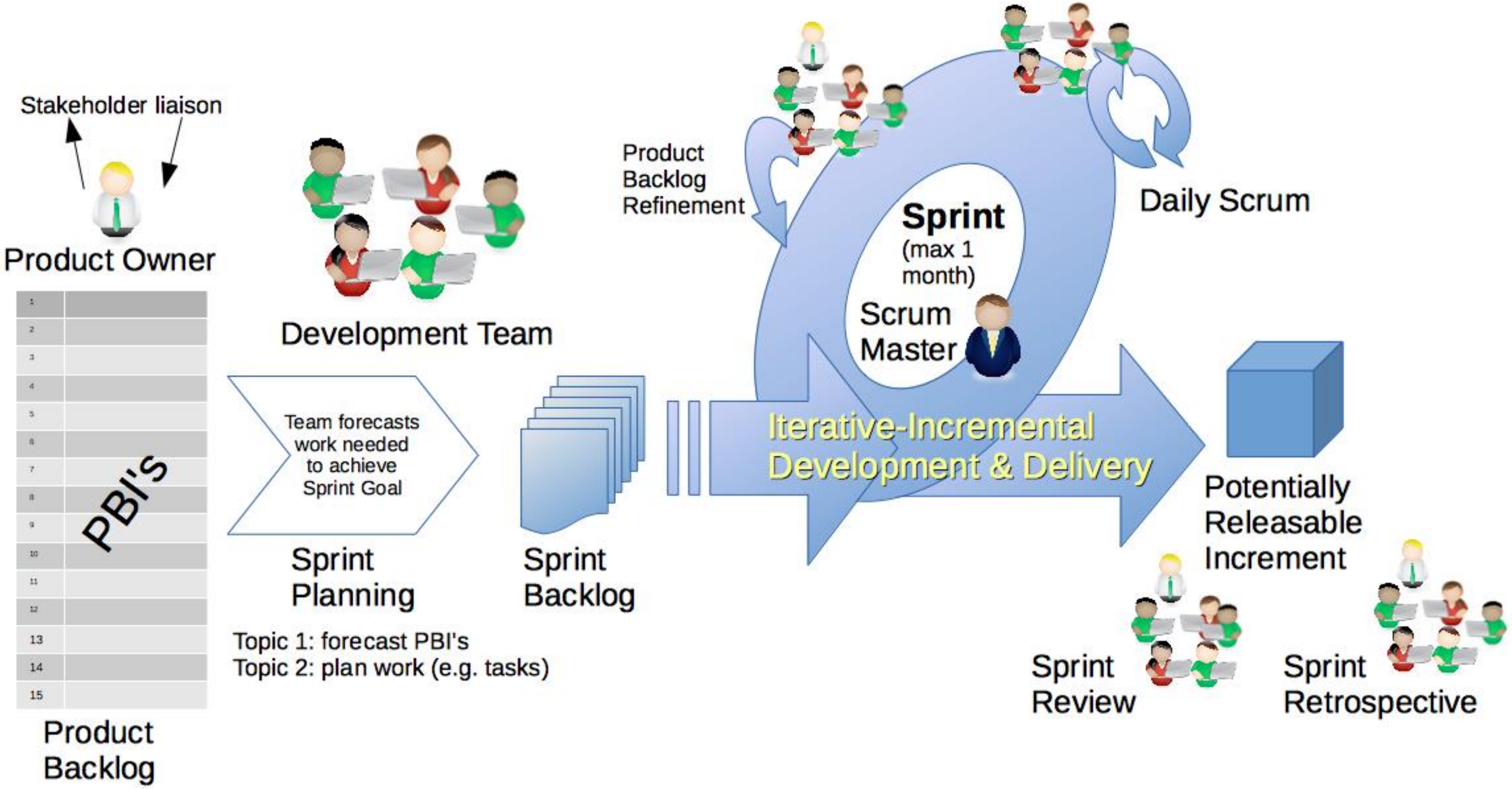
- The **Definition of Done** is a formal description of the state of the Increment when it meets the quality measures required for the product.

- The moment a Product Backlog item **meets the DoD**, an Increment is born.

- The **DoD** creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment.

- The Developers are required **to conform** to the Definition of Done.

Scrum Framework



From: Wikimedia Commons

Scrum with Remote Teams

- After pandemic, the **remote work** trend in software development became dominant, with **distributed teams** as the norm.
- Applying Scrum with distributed teams presents **challenges**, such as **communication**, **different time zones**, **isolation feelings** and project **knowledge scattering**.
- There are some points to consider in order to apply Scrum with remote teams successfully:
  - A solid communication plan
  - Effective collaboration tools
  - Informal communication channels
  - Daily scrum meetings
  - Clarifying product backlog
  - Ownership and accountability towards self-organization

Crossword

Let’s do a crossword:

https://www.scrumstudy.com/CrossWord/index.html

Useful Resources (and References)

- Agile Manifesto: https://agilemanifesto.org/
- Agile Glossary: https://www.scrumstudy.com/freeresources/scrum-agile-glossary
- Scrum Guide: https://scrumguides.org/
- Scrum.org: https://www.scrum.org/
- Scrum in Atlassian: https://www.atlassian.com/agile/scrum
- Kanban in Atlassian: https://www.atlassian.com/agile/kanban