

CSSWENG T3 AY20-21

Agile Software Development

Agile Manifesto

We value...

1. Individuals and interactions over process and tools
2. Customer collaboration over contract negotiation
3. Working software over comprehensive documentation
4. Responding to change over following a plan

Principles

1. At regular intervals, the team reflects on how to become more effective, then adjusts its behavior accordingly.
2. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

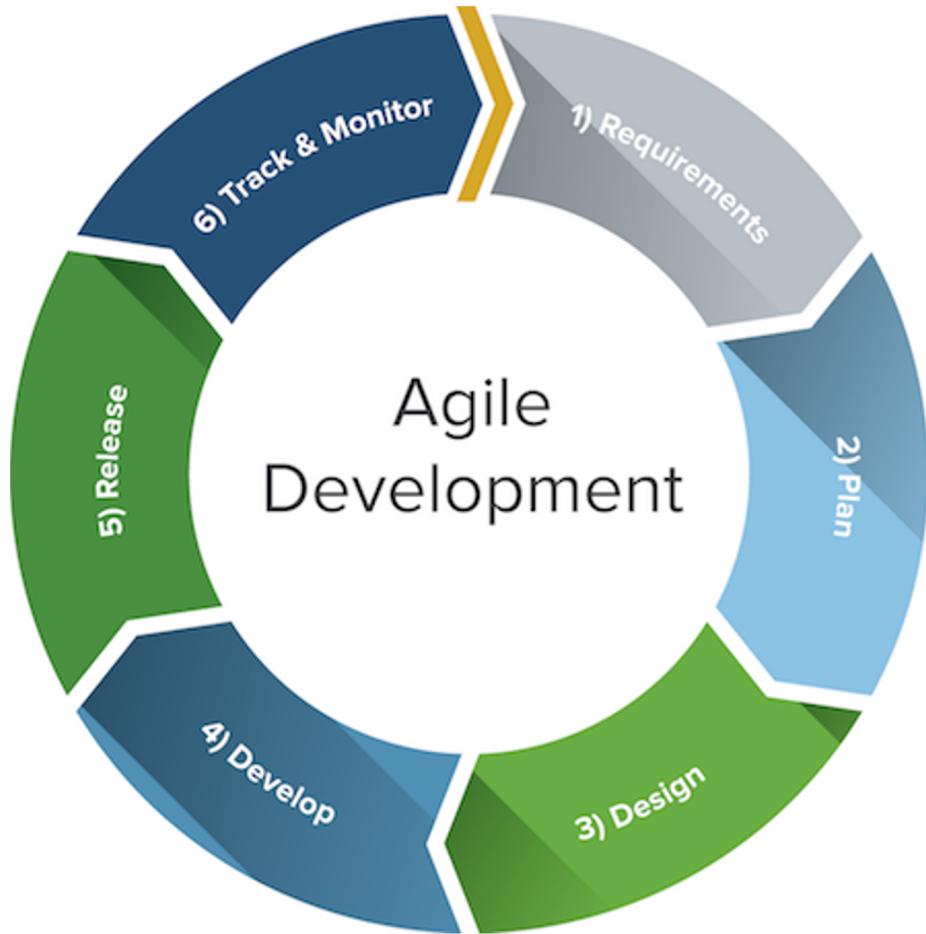
Agile Methodologies

- ✓ based on an **incremental, iterative** approach
- ✓ team leads promote **good team dynamics** through **efficient** face-to-face conversations and **strong accountability** among its members
- ✓ success is achieved through **motivated** individuals

Disadvantages of Agile

- ✗ planning is **less concrete**
- ✗ teams must be **knowledgeable** (*i.e. expertise must be of a higher than average level*)
- ✗ team must be **completely** dedicated (*as lack of dedication can cause the project to fail due to non-performing members*)

Cycle & Phases



Agile Methodologies

Kanban

Lean Software Development

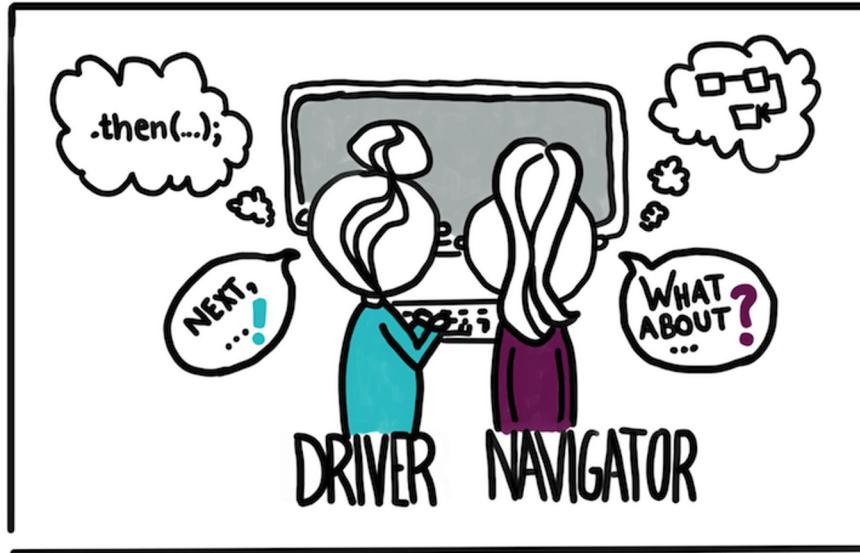
Feature-Driven Development

Adaptive System Development

Crystal Clear

Extreme Programming (XP)

Scrum



Agile & Pair Programming

Photo from Martin Fowler

Pair Programming

- supports the idea of collective ownership
- an informal review process
- encourages refactoring

Roles

1. Driver
2. Navigator

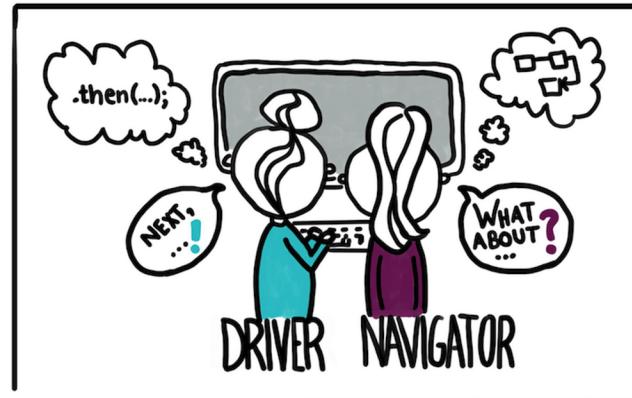


Photo from Martin Fowler

Pair Programming Approaches

How to pair

- Driver and Navigator
- Ping Pong
- Strong-Style Pairing
- Pair Development

To know more, read [On Pair Programming | Martin Fowler](#)

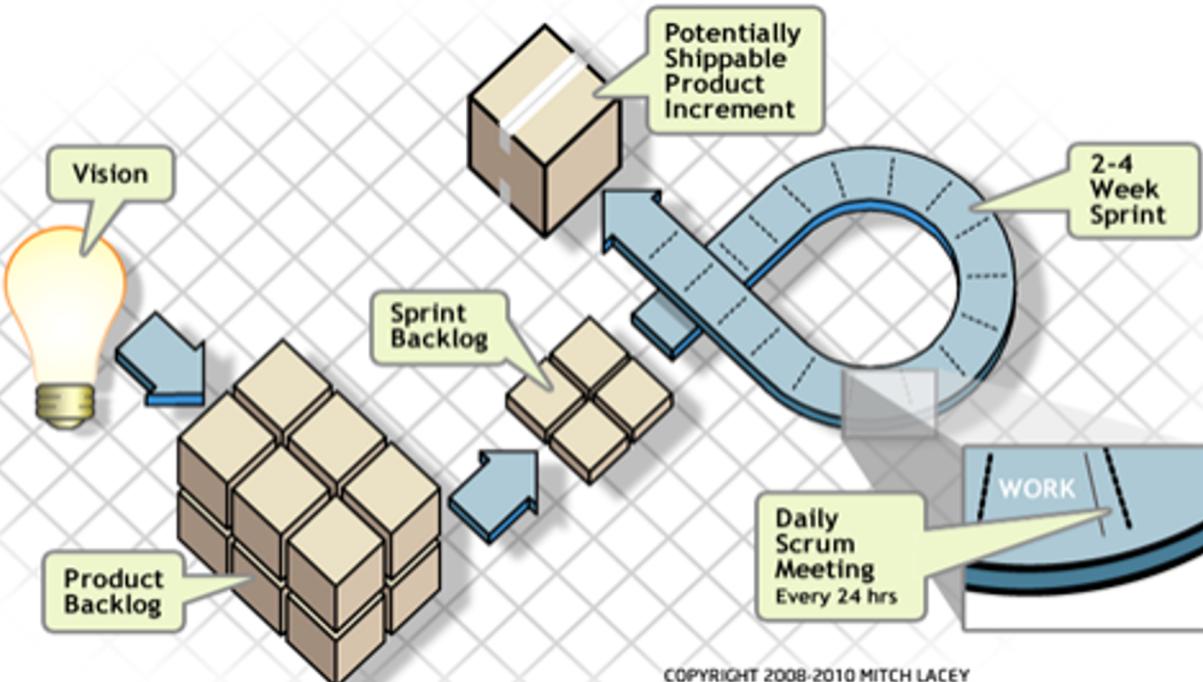


SCRUM

Scrum Agile Method

A framework for organizing agile projects

Provides external visibility of what is going on



Scrum Theory

Transparency

- Significant aspects of the process must be visible to those responsible for the outcome
- A common standard must be defined for a common understanding

Inspection

- Members must inspect Scrum artifacts and progress towards the Sprint Goal

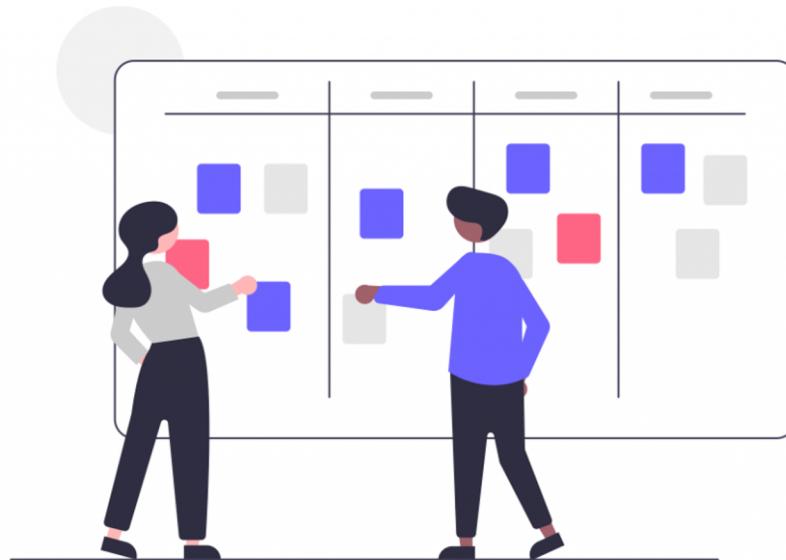
Adaptation

- Adjustments can be made to minimize deviation from the plan

Scrum Team

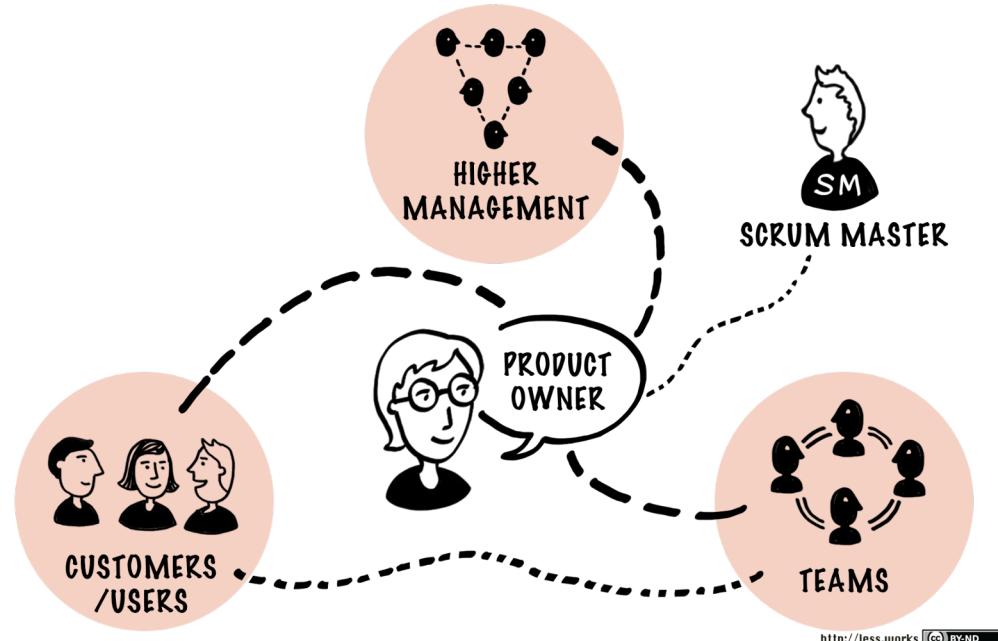
A Scrum Team is composed of the following roles, but the people assigned to these roles are not fixed.

- Product Owner
- Scrum Master
- Developers



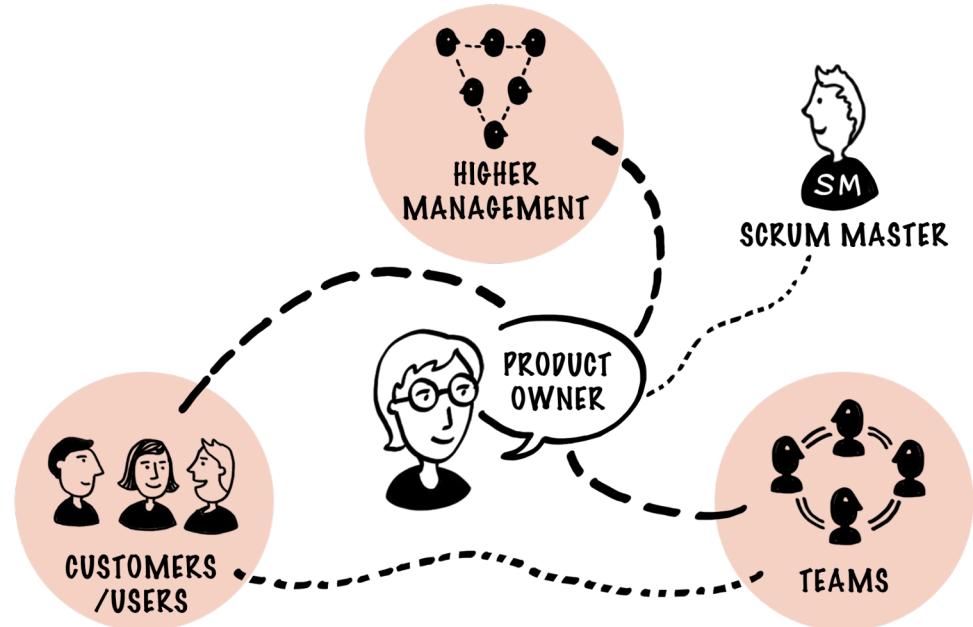
Scrum Team - Product Owner

- The product owner is one person, not a committee
- Represents the stakeholders and the best interest of the customers
- Responsible for delivering good business results
- Maximizes the value of the product that the team delivers



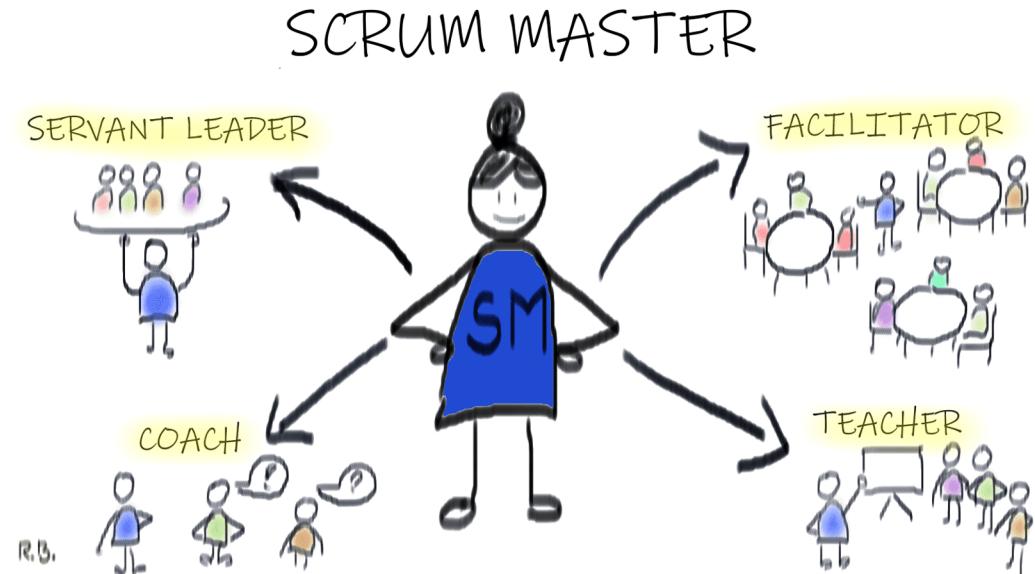
Scrum Team - Product Owner

- Manages the product backlog
- Prioritizes the items in the backlog based on importance and dependencies
- Focuses on the business side of product development



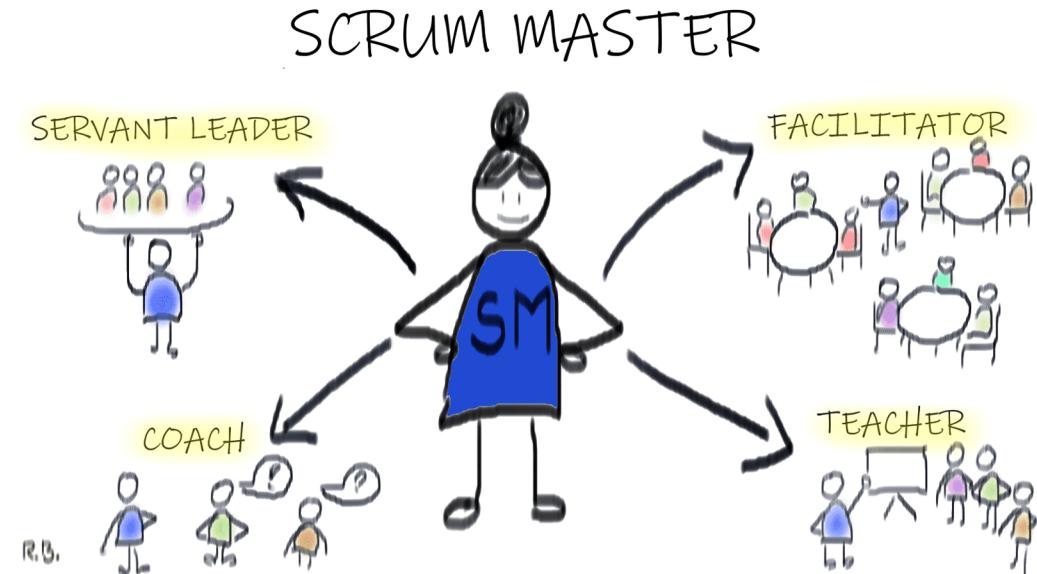
Scrum Team - Scrum Master

- Similar to a project manager
- Accountable for removing impediments that hinder the team in delivering the product goals and deliverables



Scrum Team - Scrum Master

- Coaches and ensures that the team adheres to Scrum theory, practices and rules
- Coordinates with the product owner in maintaining the product backlog so that the team understands the remaining work to be done



Scrum Team - Developers

- Carry out all the work required to build increments of value every sprint
- Create a releasable increment of the “Done” product, having a size that should be small enough to remain nimble but large enough to complete work
- The developers are cross-functional. As a team they have the skills necessary to create a product increment.
- Accountability belongs to the team as a whole (even when each developer has a specific area of focus)

Scrum Team - Designer & Quality Assurance

- The designer and quality assurance (QA) roles are not distinguished in Scrum.
- Designers work on the designing and even developing the user interface and overall user experience.
- The product owner takes up most of the designer role.
- QAs develop test plans and automate tests to efficiently assess the correctness and validity of the software solution.
- The QA team is part of the development team.

Scrum Events

Sprint Planning

Setting of Sprint Goals at the start of a Sprint

Daily Scrum

A 15-minute time-boxed meeting at the start of every day. Activities to be done within the day are planned in the daily scrum.

Sprint Review

Performed at the end of every Sprint to discuss what was done during the Sprint and to inspect the Product Backlog

Sprint Retrospective

Opportunity for the team to reflect on improvements at the end of every Sprint

SPRINT

The heart of Scrum

It is a time-box of one month or less during which a “Done”, usable, and potentially shippable product increment is created.

Artifacts

Product Backlog

An ordered list of everything that is needed for the project, and is a single source of project requirements. The Product Owner is responsible for maintaining this.

Sprint Backlog

A set of Product Backlog items selected for the Sprint, plus a plan for realizing the Sprint Goal

Increment

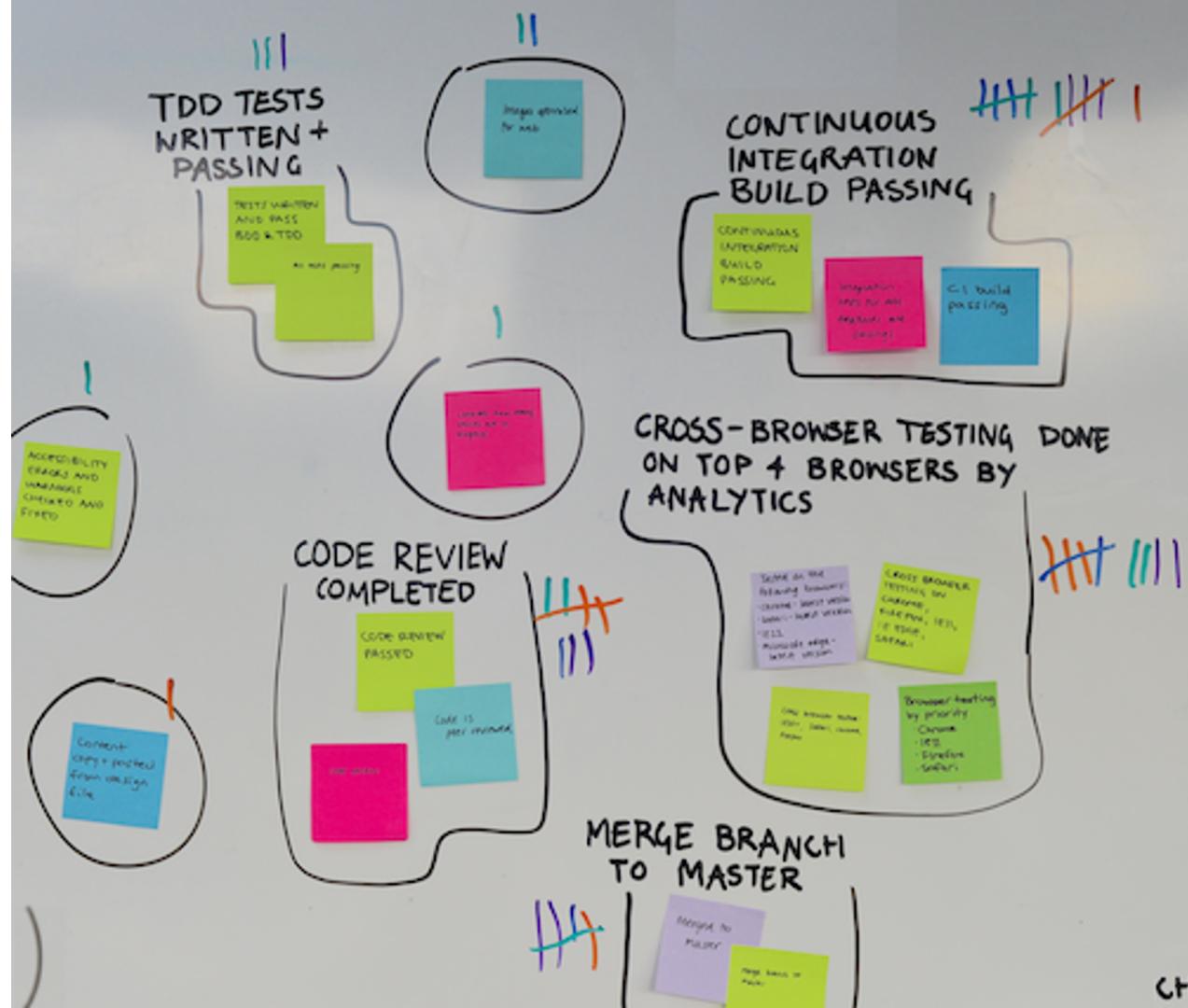
The Increment is the sum of all Product Backlog items completed during a sprint and the value of the increments of all previous Sprints.

Definition of Done

- Scrum relies heavily on artifact transparency
- Everyone must have shared understanding of what “Done” is
- Defined by the team
- This will guide the Scrum Team in selecting Product Backlog items during Sprint Planning
- Need for Stories, Sprints, Integration, and Releases

Definition of Done

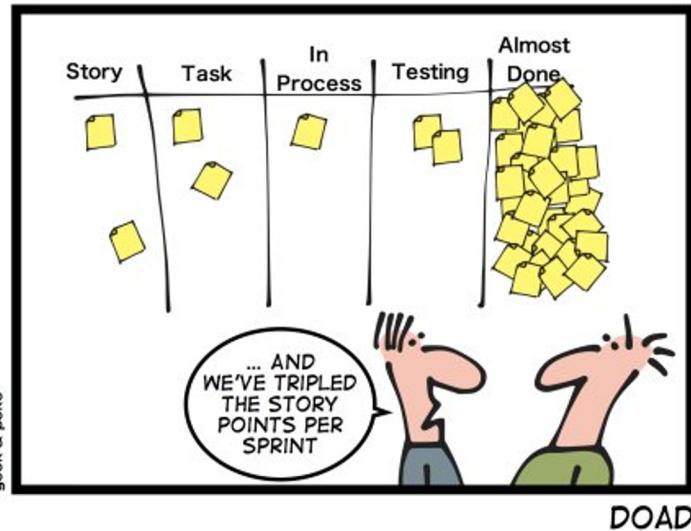
Checklist of things to be met by the user story prior to moving the task to “Done”



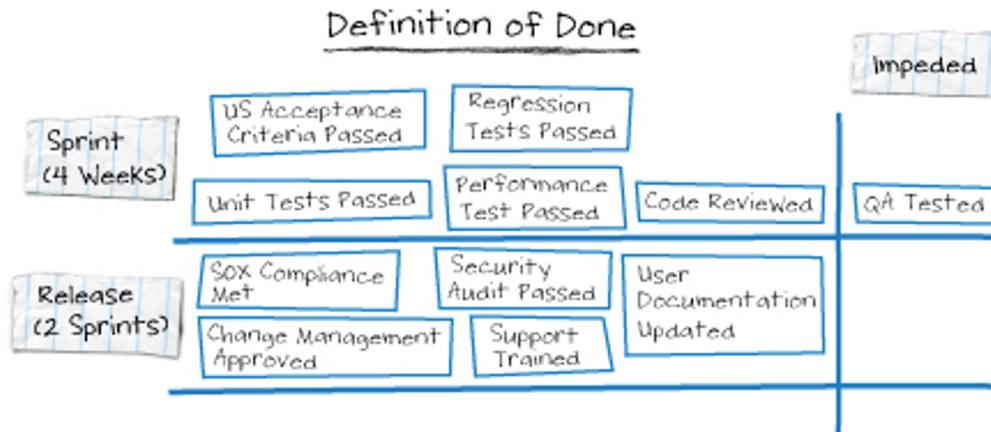
Definition of Done (DoD)

The below examples might be included in a User Story DoD:

- Unit tests passed
- Code reviewed
- Acceptance criteria met
- Functional Tests passed
- Non-Functional requirements met
- Product Owner accepts the User Story



DOAD



Reference

The 2020 Scrum Guide – <https://scrumguides.org/scrum-guide.html>



Questions?