Agile

Definition

- Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.

- represented the adaptiveness and response to change
- https://martinfowler.com/articles/agileStory.html

More then a method

- Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it.

Focus

- focus on the people doing the work and how they work together
- Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context.

Manifesto

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.

http://agilemanifesto.org/

Principles

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Business people and developers must work together daily throughout the project.
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Working software is the primary measure of progress.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Continuous attention to technical excellence and good design enhances agility.	Simplicity--the art of maximizing the amount of work not done--is essential.	The best architectures, requirements, and designs emerge from self-organizing teams.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

http://agilemanifesto.org/principles.html

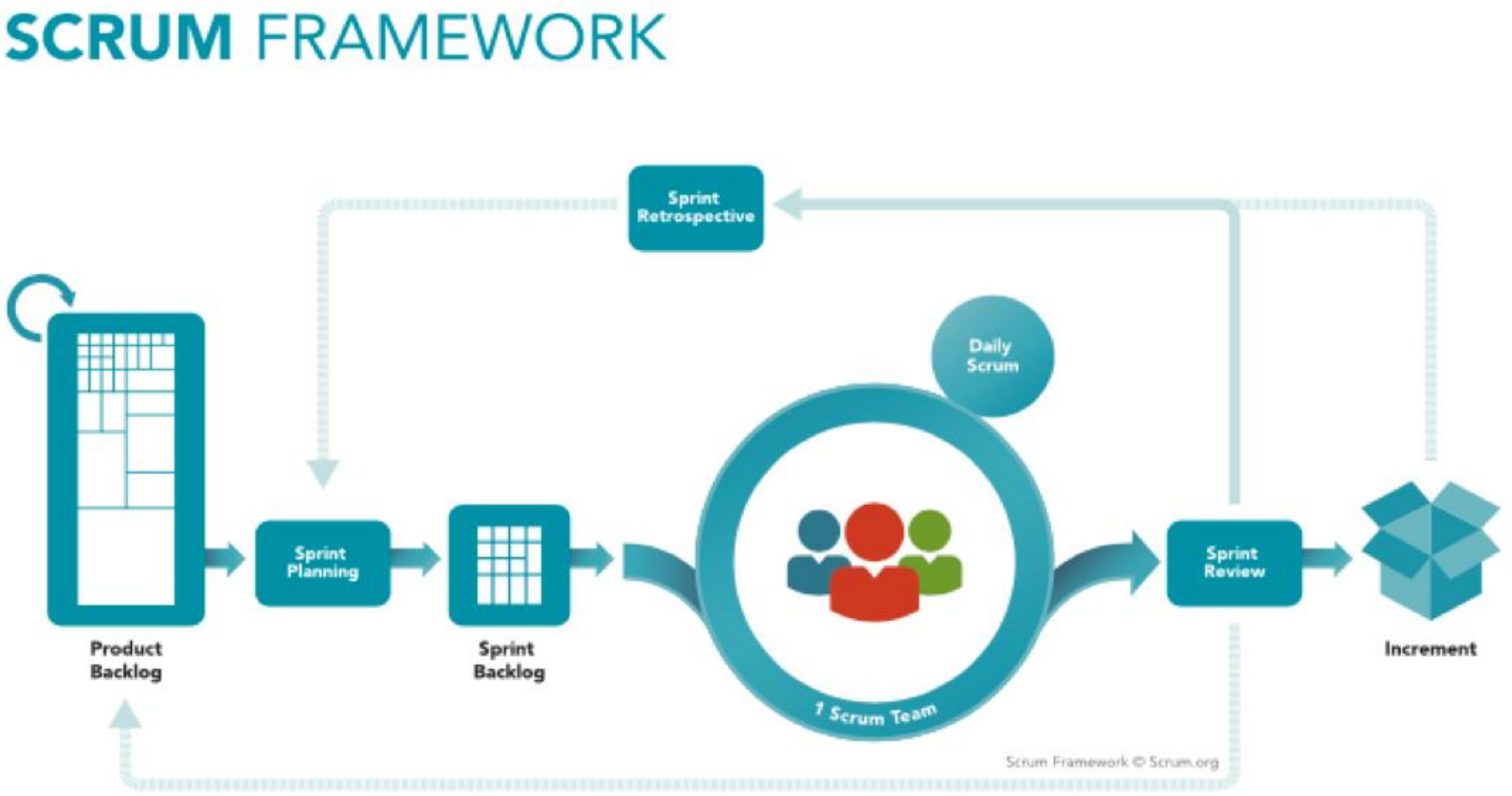
Scrum

- developing, delivering, and sustaining complex products.

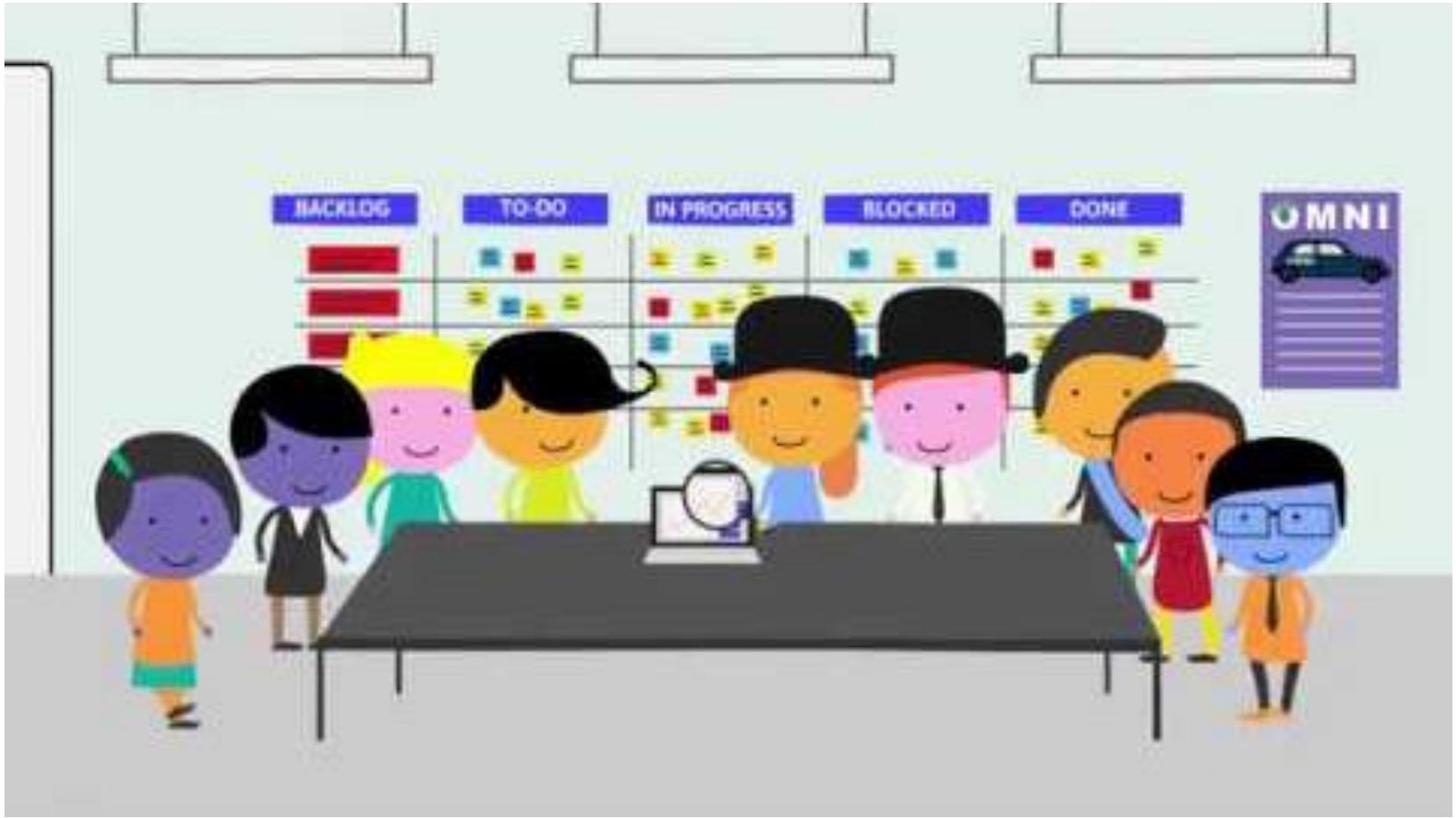
Definition

- A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

The Scrum Framework



An Introduction to the Scrum Framework



The Scrum Values

The Roles of the Scrum Team

- Product Owner
- Development Team
- Scrum Master

- Self-organizing
  - teams choose how best to accomplish their work, rather than being directed by others outside the team.
- Cross-functional
  - teams have all competencies needed to accomplish the work without depending on others not part of the team.

- optimize flexibility, creativity, and productivity.

The Scrum Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

- All events are time-boxed

Scrum Artifacts

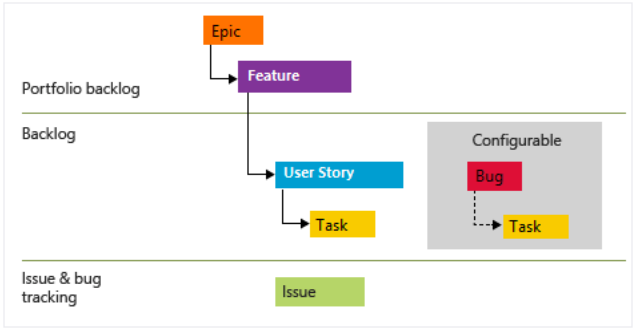
- Product Backlog
- Sprint Backlog
- Increment

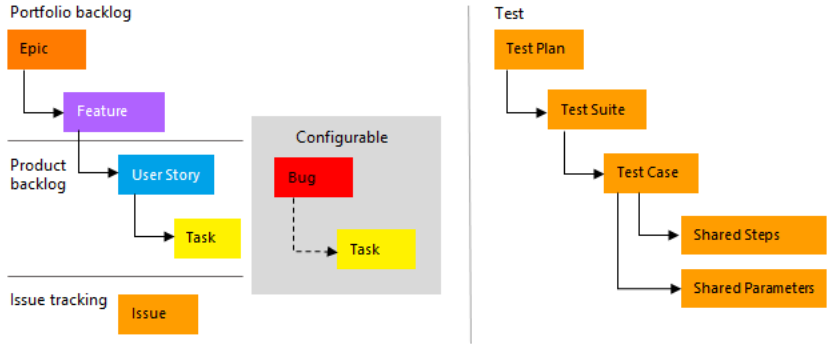
- artifacts represent work or value to provide **transparency** and opportunities for inspection and adaptation

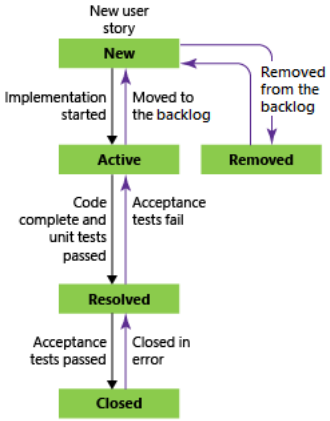
Resources

- https://www.scrum.org/resources/what-is-scrum
- http://www.scrumguides.org/

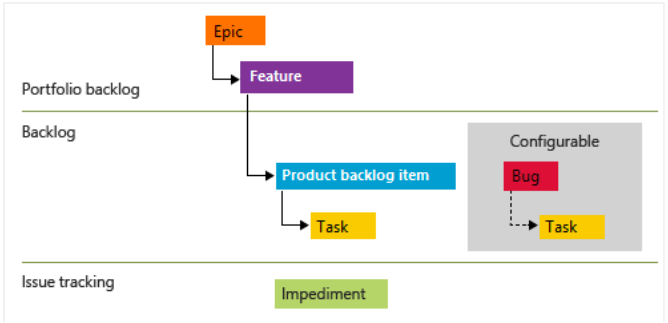
Agile

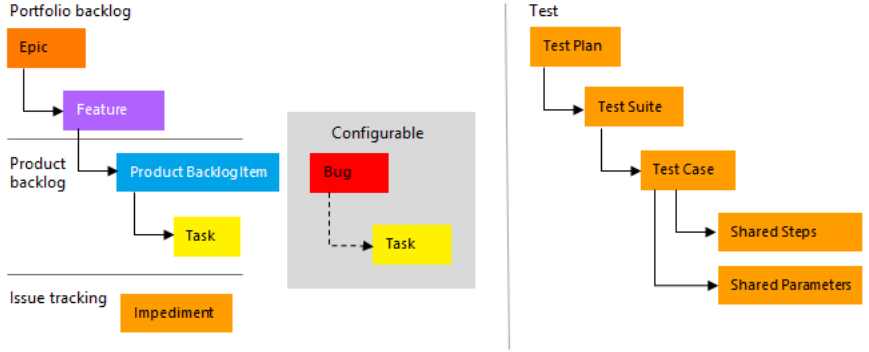

Epic
Feature
Portfolio backlog
Backlog
Configurable
User Story
Bug
Task
Task
Issue & bug tracking
Issue

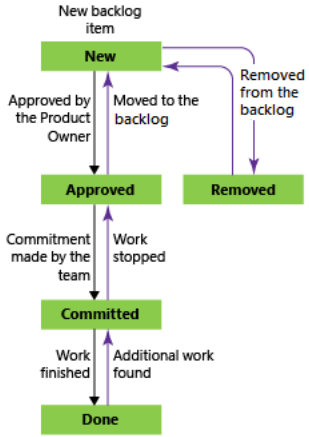

Portfolio backlog
Epic
Feature
Product backlog
User Story
Task
Configurable
Bug
Task
Issue tracking
Issue
Test
Test Plan
Test Suite
Test Case
Shared Steps
Shared Parameters


New user story
New
Implementation started
Moved to the backlog
Removed from the backlog
Active
Removed
Code complete and unit tests passed
Acceptance tests fail
Resolved
Acceptance tests passed
Closed in error
Closed

Scrum







Define stories, requirements

- Create your backlog
- Prioritize your backlog
- Estimate work
- Assign work

Manage portfolios

- Add epics or features
- Group backlog items

Manage bugs

- Capture bugs
- Triage bugs
- Resolve & close bugs

Manage issues

- Capture issues
- Track dependencies
- Query issues

Differences Agile - Scrum

Tracking area	Agile	Scrum
Workflow states	•New •Active •Resolved •Closed •Removed	•New •Approved •Committed •Done •Removed
Product planning (see note 1)	•User story •Bug (optional)	•Product backlog item •Bug (optional)
Portfolio backlogs (2)	•Epic •Feature	•Epic •Feature
Task and sprint planning (3)	•Task •Bug (optional)	•Task •Bug (optional)
Bug backlog management (1)	•Bug	•Bug
Issue and risk management	•Issue	•Impediment