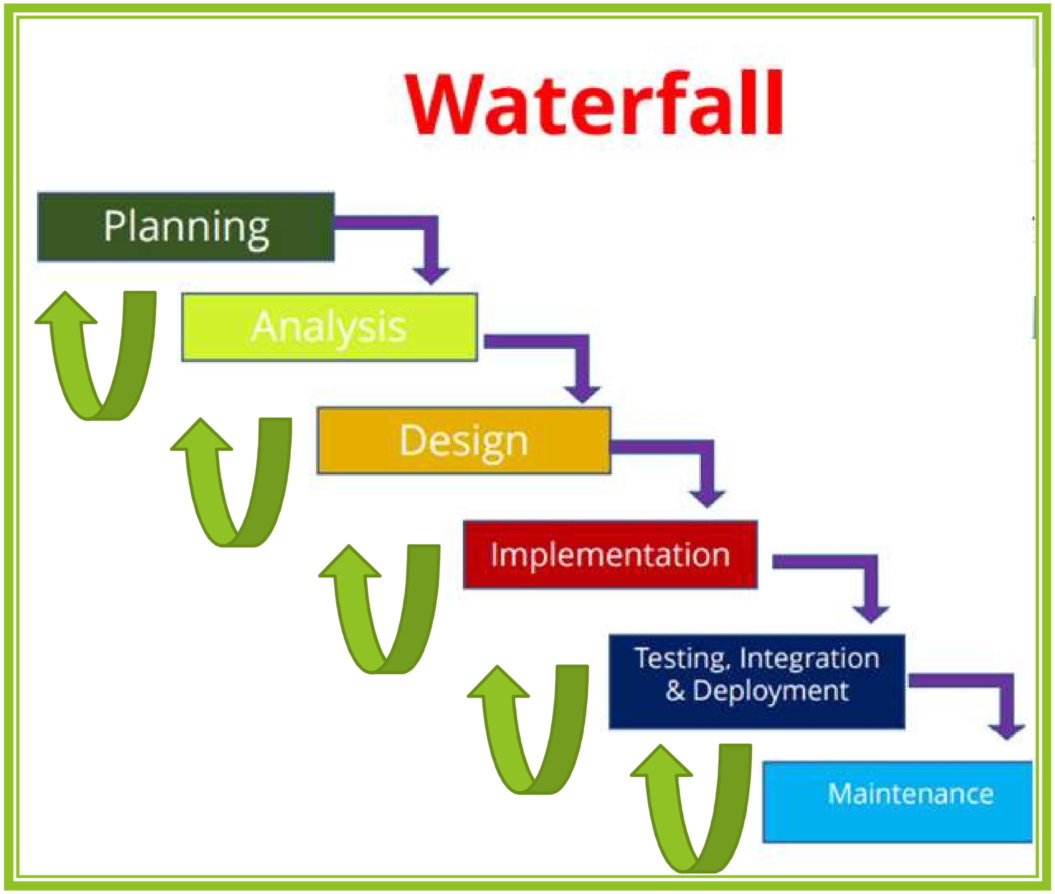
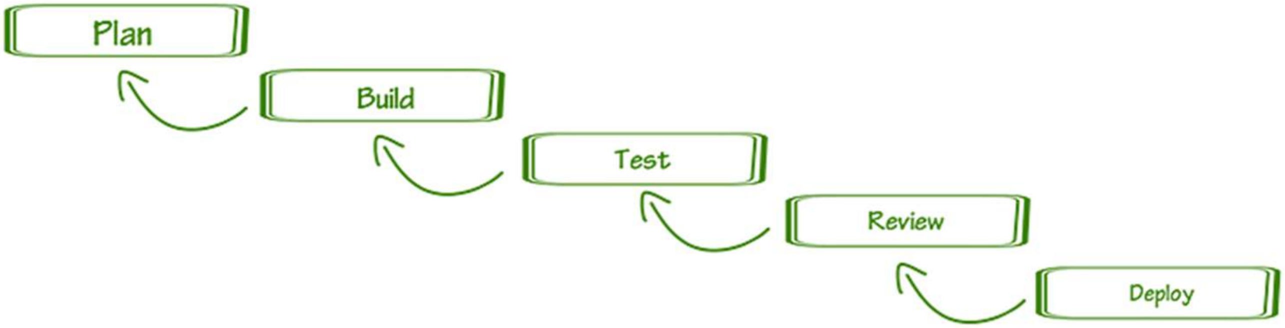
Software Project Management

AGILE METHOD

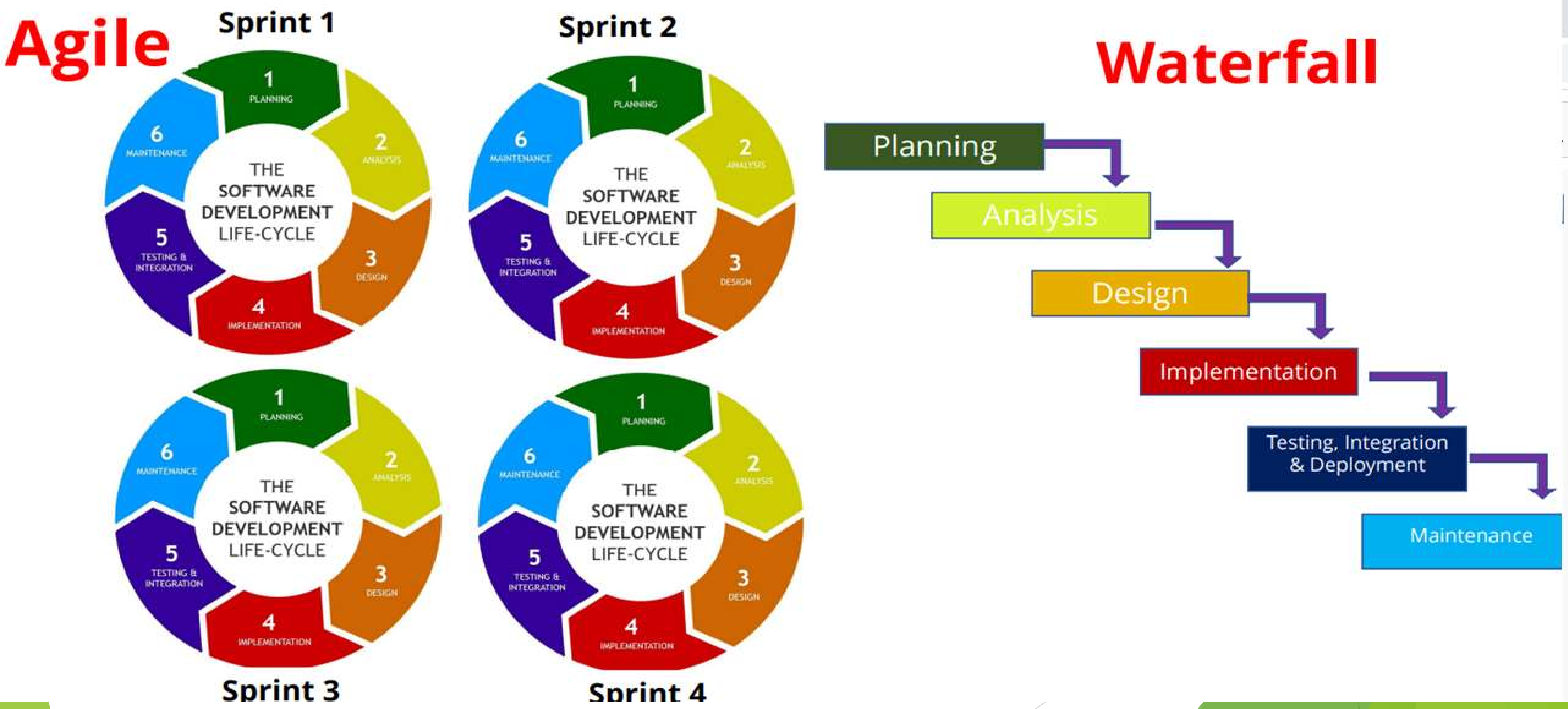
Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in an uncertain environment. Agile is an iterative approach to project management and software development that help teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly


Waterfall
Planning
Analysis
Design
Implementation
Testing, Integration & Deployment
Maintenance

Waterfall



Agile vs Waterfall



The Agile Manifesto:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile - 12 Principles

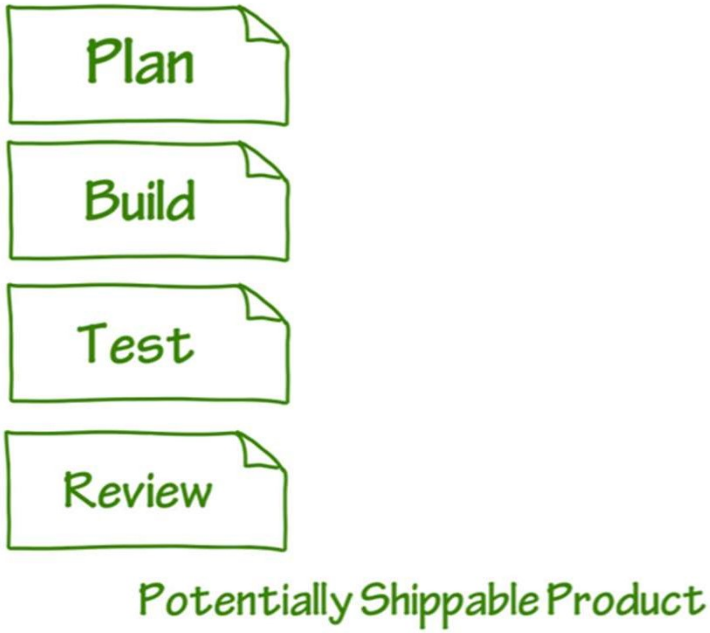
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of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

Agile Methodologies

1. Scrum
2. Kanban
3. Extreme Programming (XP)
4. Feature -driven development (FDD)
5. Dynamic Systems Development Method (DSDM)
6. Crystal
7. Lean

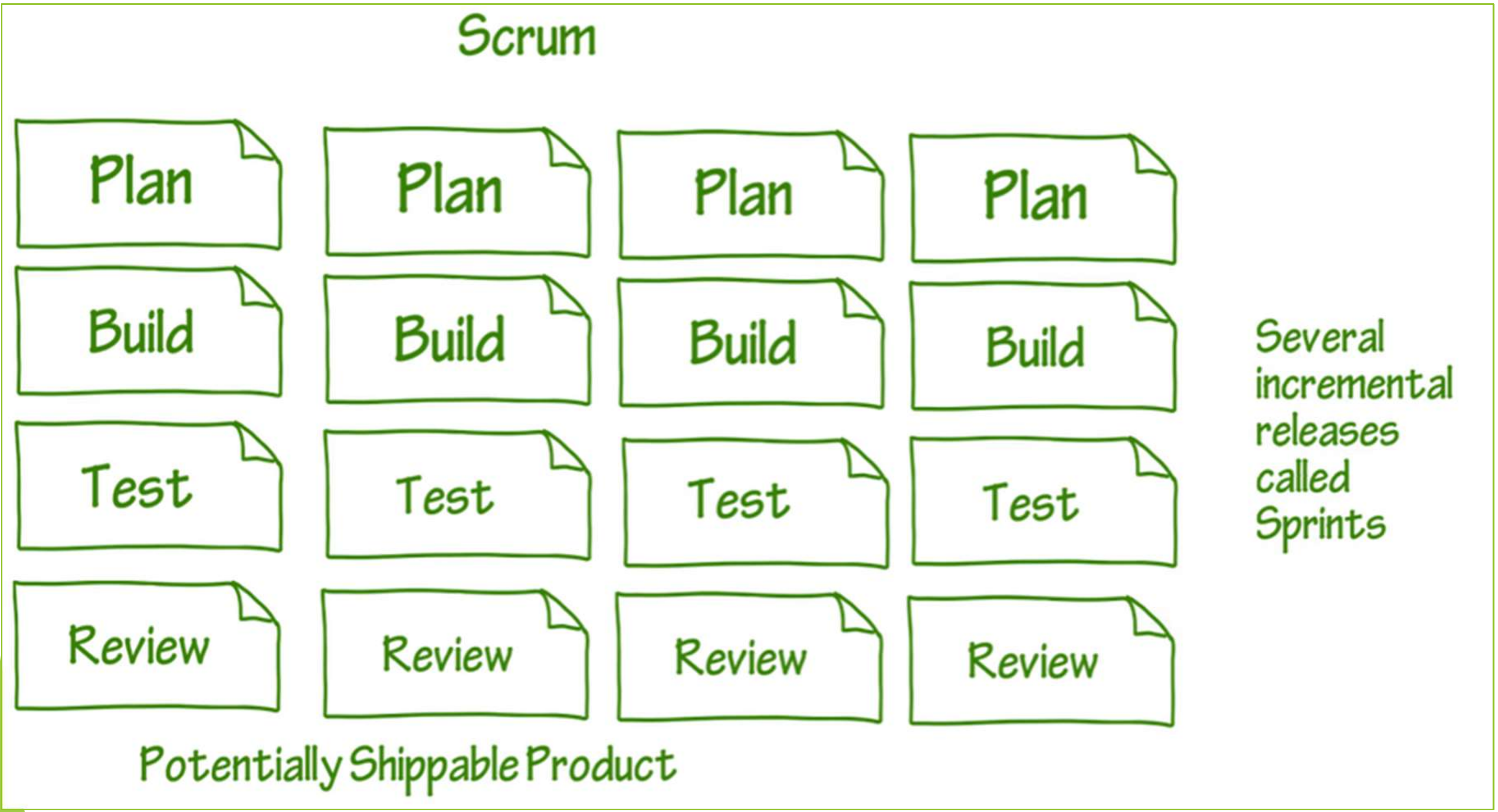
SCRUM

STEP 1

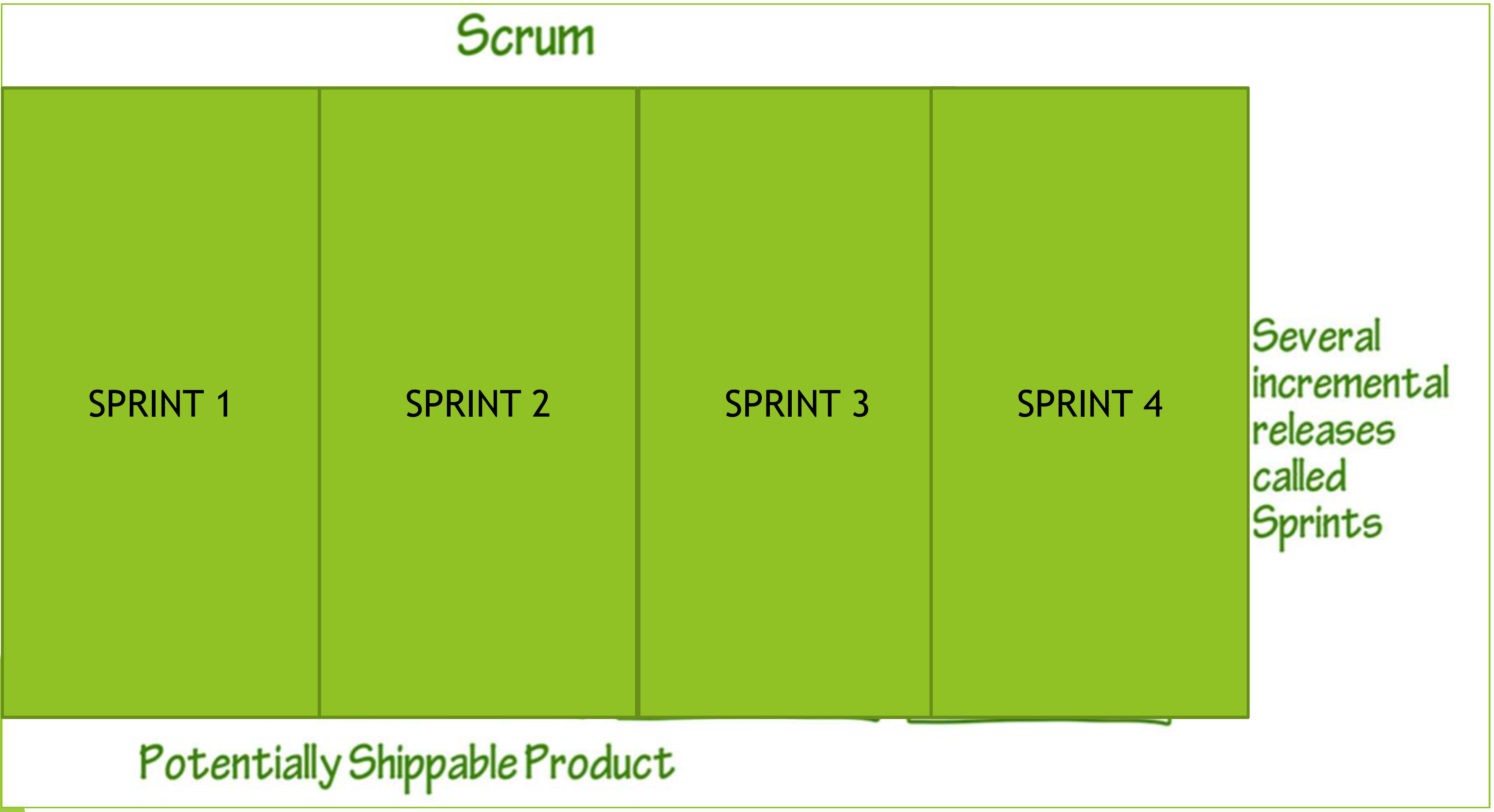


- With SCRUM, an implementation of an agile, used to manage software project development, the process is broken up into smaller pieces.
- First we do just enough planning to get started with, building a minimal feature set.
- We build what was planned, next we test and review that small feature set and get it ready to a ship.
- When that cycle is complete we end up with a potentially shippable product
- This process usually occurs in a time period of 1 to 3 weeks.
- This is then repeated with time to time

Scrum is best suited in the case where a cross functional team is working in a product development setting where work can be split into 1 – 4 weeks iterations called Sprints.



- Repeat the cycle, time and time again
- Reducing the time from planning to development to testing and reviewing
- Each time to the planning process, doing enough planning to complete the next incremental release
- You end up with several incremental releases called sprints



Scrum

SPRINT 1

SPRINT 2

SPRINT 3

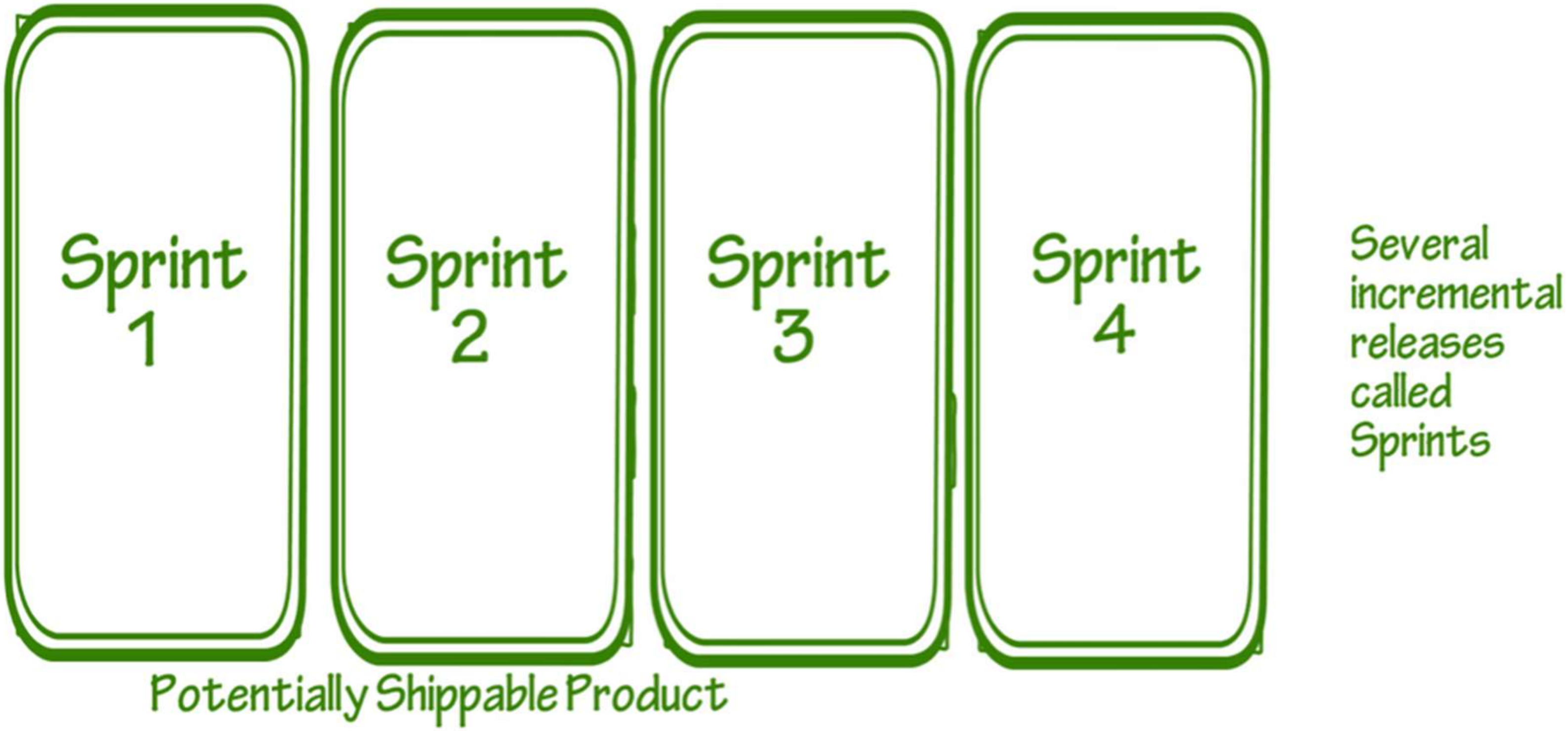
SPRINT 4

Several incremental releases called Sprints

Potentially Shippable Product

- ➢ Repeat the cycle, time and time again
- ➢ Reducing the time from planning to development to testing and reviewing
- ➢ Each time to the planning process, doing enough planning to complete the next incremental release
- ➢ You end up with several incremental releases called sprints

SCRUM


Sprint 1
Sprint 2
Sprint 3
Sprint 4
Several incremental releases called Sprints
Potentially Shippable Product

- A Sprint usually takes 1 to 3 weeks and you just keep repeating the sprints until your product is feature complete
- Sometimes you may end up shipping the product after the second sprint or the 3rd or the 4th but you eventually end up with the potentially shippable product

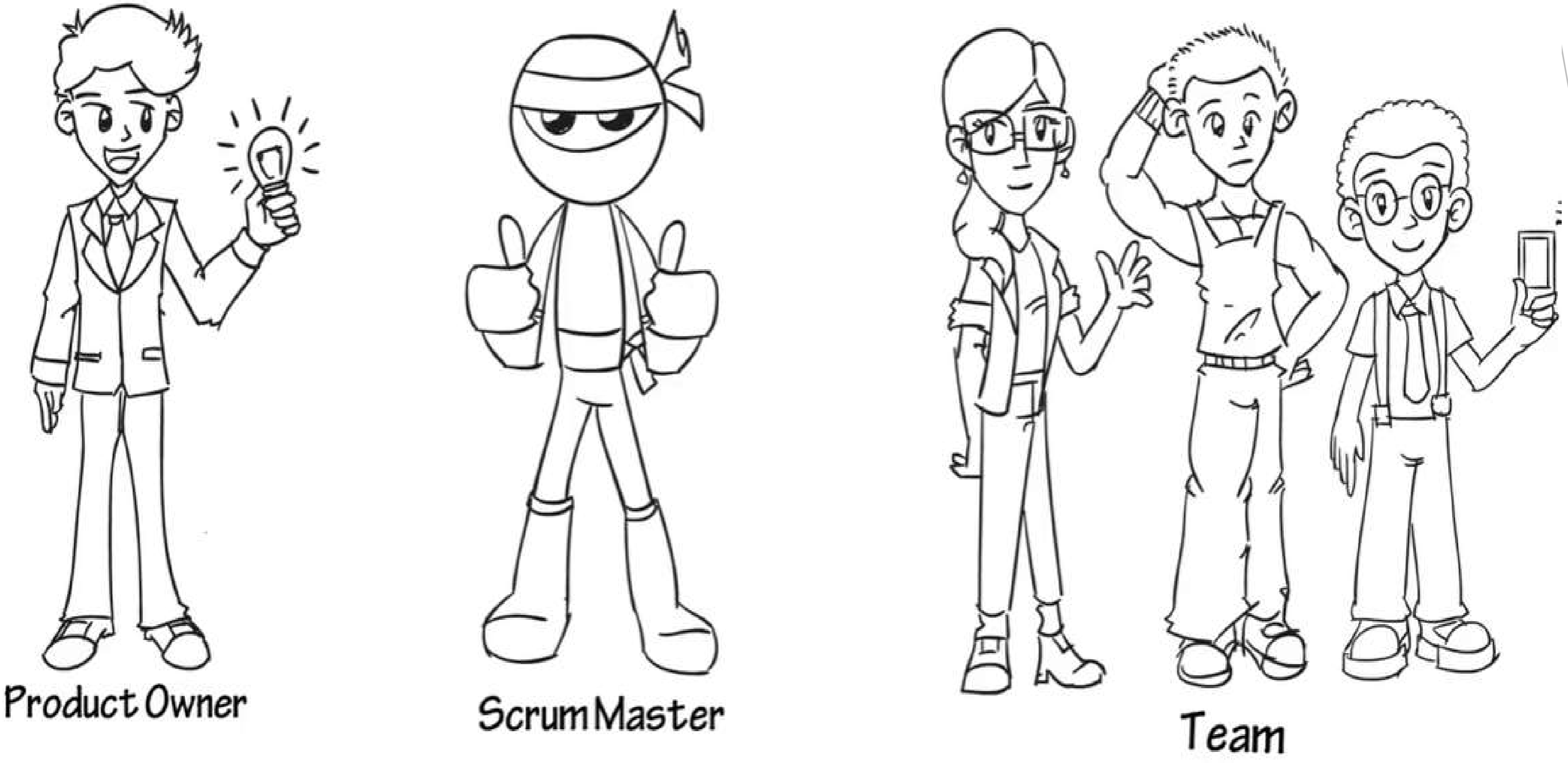
SPRINT

The Sprint is a time box of one month or less during which the team produces a potentially shippable product Increment.

Typical characteristics of Sprints:

- Maintain a consistent duration throughout a development effort
- A new Sprint immediately follows the conclusion of the previous Sprint
- Start date and end date of Sprint are fixed

SCRUM Roles



3 key roles that are needed for the framework to work well

Scrum Roles

- **The Product Owner**

The Product owner is the person responsible for defining the feature that are needed for the product.

Product owner has bright ideas that turn into products.

The product owner manages the product backlog and reviews the sprint output.

- **The Scrum Master**

The Scrum master is a leader to the team responsible to protect the team in the process, running the meetings and keeping things going

The scrum master is responsible for ensuring the team lives Agile values and principles and follows the processes and practices that the team agreed they would use. Scrum Master leads organizes and leads Scrum Ceremonies.

- **The Development Team**

Team can be made up of developers, testers, riders and anyone else that helps in developing the product. Team members often play multiple roles, sometimes developers may end up doing test

The development team consists of the people who deliver the product increment inside a Sprint. The main responsibility of the development team is to deliver the increment that delivers value every Sprint.

ARTIFACTS

There are 3 artifacts or documents that are used in SCRUM

Product Backlog

- Product owners create a prioritize list of features known as user stories that could go on the product.
- This list evolves and changes with every sprint
- The product backlog is an ordered list of all the possible changes that could be made to the product. Items on the product backlog are options, not commitments
- Product Backlog does not guarantee they will be delivered.

User Stories

- User stories are a way of describing feature set, that falls the AS A user, I need something so that reason format. This kind of phrasing helps the product owner to specify the right amount of detail for the team to estimate the size of task.
- Also called "user stories," are short requirements or requests written from the perspective of an end user.

Epics

Epics are large bodies of work that can be broken down into a number of smaller tasks (called stories).

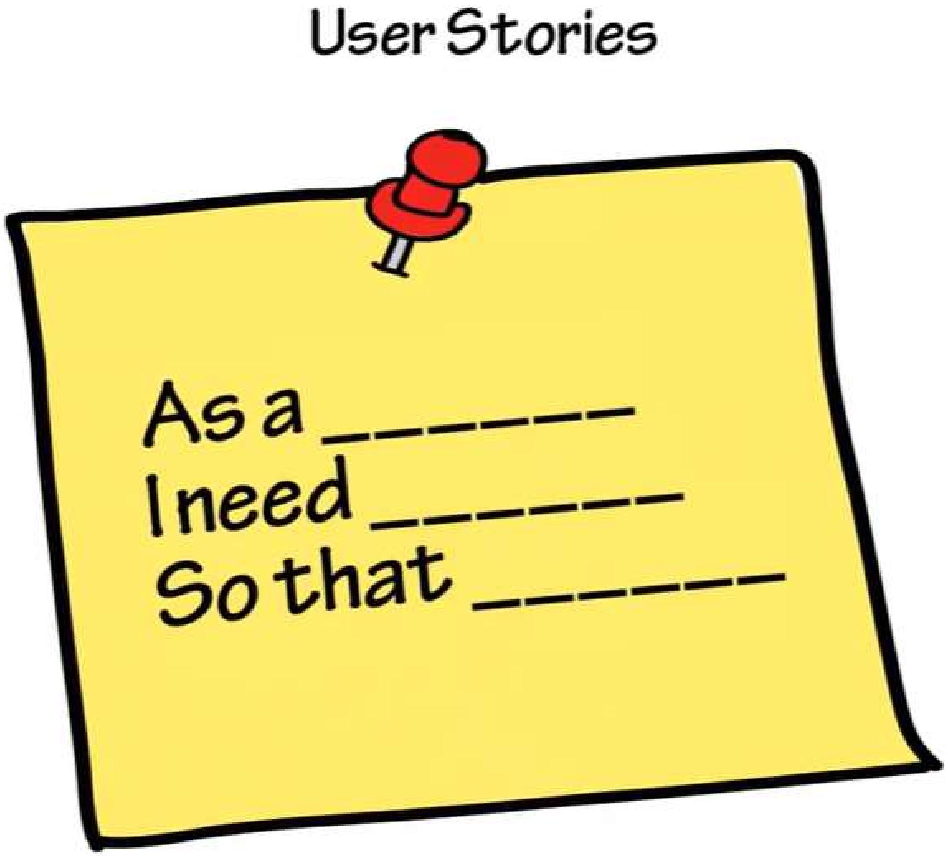
Sprint Backlog

The highest priority user stories go into the sprint backlog, estimated for the size and are committed to for the next sprint

The Sprint Backlog is the collection of product backlog items(user stories) selected for delivery in the Sprint.

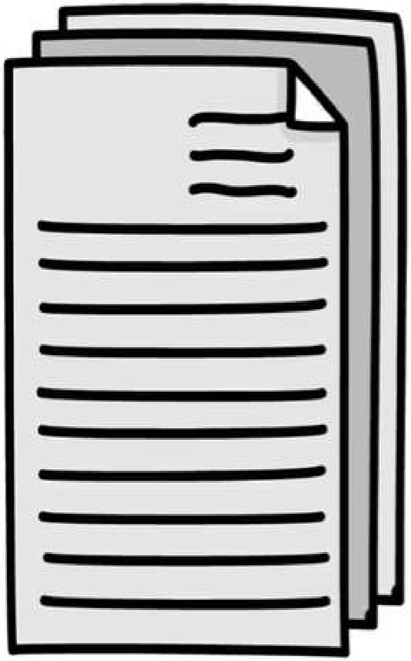
Burndown chart shows the progress on the completion of tasks in the sprint backlog

ARTIFACTS

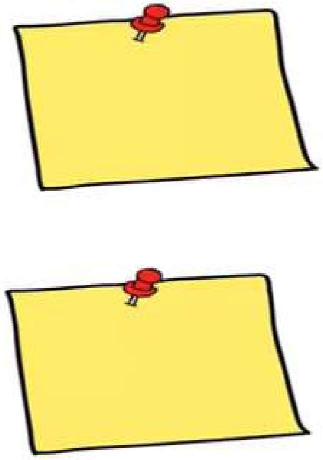


Artifacts

Product Backlog



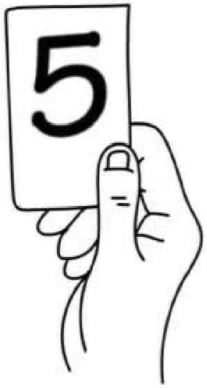
Sprint Backlog



Burndown Chart

Scrum Ceremonies

Sprint Planning



Daily Scrum

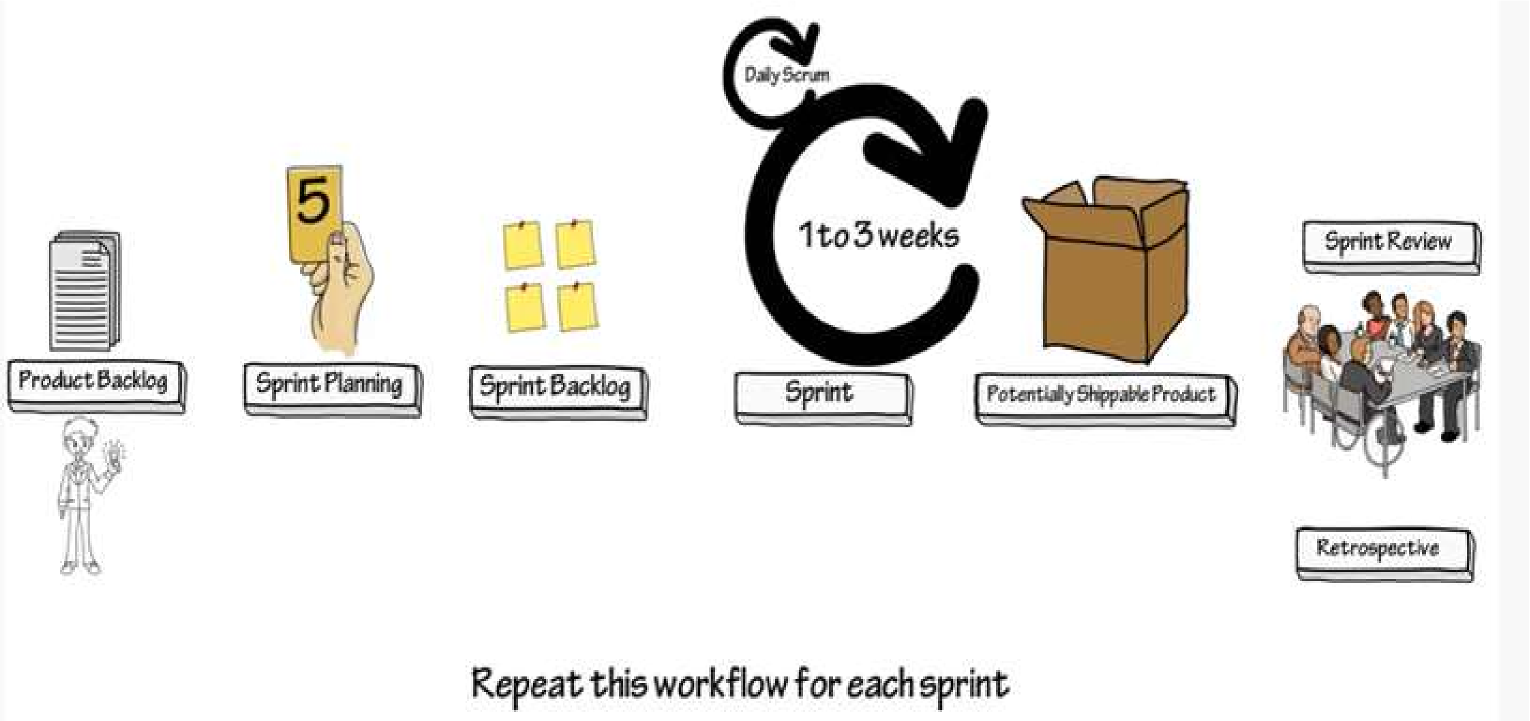
Sprint Review

SCRUM CEREMONIES

There are 3 ceremonies that makeup scum. Think of these as meetings or discussions

- Sprint planning is where the product owner, scrum master and team discuss about the user stories and estimate the relative sizes
- Daily scrum is a brief standup meeting where the team meets to discuss the progress, what they are working on or anything that is blocked or require help.
- Sprint review and retrospective occurs at the end of the sprint. This is where the team demonstrates the completed work to the product owner and then the team discusses what they can do to improve the process going forward.

SCRUM WORKFLOW


Daily Scrum
5
1 to 3 weeks
Product Backlog
Sprint Planning
Sprint Backlog
Sprint
Potentially Shippable Product
Sprint Review
Retrospective
Repeat this workflow for each sprint