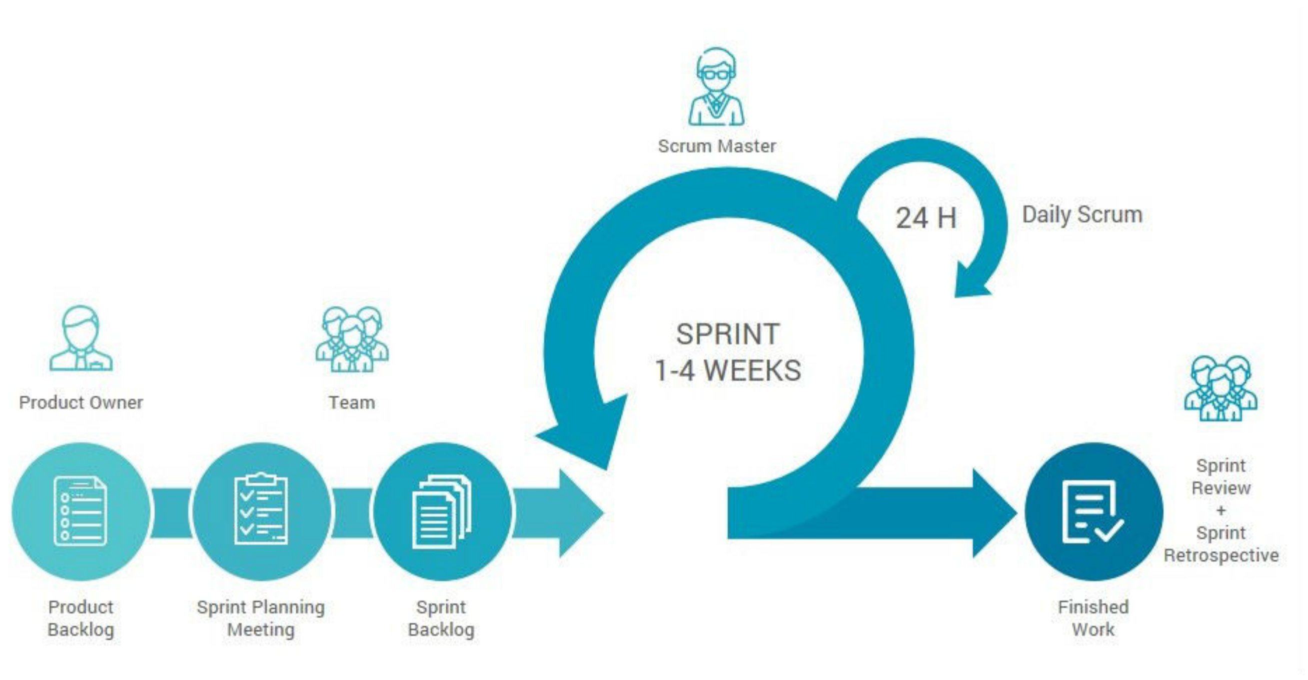
SD Methodologies III

1 SCRUM

1.1 Sprint



- The starting point for planning is the product backlog (user stories), which is the list of work to be done on the project
- Once these are agreed, the team organise themselves to develop the software - the team creates a spring backlog (their to-do list for the sprint). At the end of the sprint, the work done is reviewed and presented to stakeholder. The next sprint cycle then begins

1.2 Artifacts

The product backlog:

- The scrum product backlog is simply a list of all things that need to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories
- Each scrum product backlog has certain properties that differentiate it from a simple to-do list:
  - An entry in the SPB always add value for the customer
  - The entries in the SPB are prioritized and ordered accordingly
  - All entries are estimated
  - The SPB is a living document
  - There are no action-items or low-level tasks in the SPB
  - SPB entries can be scenarios or use cases

1.3 SPB items

- What you have agreed to do (SPBI) in the current sprint; has an end
- How is breaking the SPBIs into small pieces - sprint tasks
- These tasks can be tested

1.4 Meetings

Sprint planning:

- Plan one sprint. PO and team decide what SPBIs go into sprint backlog (have top priorities); try to get it clear what they have to do and how much they can do in the time allocated (timebox)
- Use story points - arbitrary measure of effort required

Daily scrum:

- Meet daily; 15 minute stand up and report to each other NOT to the PO or SM
- What you did yesterday, what you will do today, what problems have you got

Sprint review:

- Demo potentially shippable product increment to PO and anyone interested. PO declares what is complete and what doesn't satisfy user requirements

Retrospective meeting:

- At the end of every sprint what went well, what can be improved, what you have learned, feedback to each other - teams take ownership of the process

Refinement meeting:

- Team and PO look ahead a little at the PBI's to determine possible candidates and maybe break some down for the next couple of sprints. This can also be done in the sprint planning meeting

1.5 Estimation

How much effort is involved:

- Focus is on the collective effort not the individual effort per task
- Teams can compare items to each other or estimate in relative units
- Teams breaks it into the smallest possible stories

Velocity:

- Measure of work completed by the development team in a given sprint
- Is based on your estimated targets

1.6 Advantages

- The product is broken down into a set of manageable and understandable chunks
- Unstable requirements do not hold up progress
- The whole team have visibility of everything and consequently team communication is improved
- Customers see on-time delivery of increments and gain feedback on how the product works
- Trust between customers and developers is established and a positive culture is created in which everyone expects the project to succeed

1.7 Disadvantages

- It can be difficult to keep the interest of customers who are involved in the process
- Prioritising changes can be difficult where there are multiple stakeholders
- Maintaining simplicity requires extra work
- Contracts may be a problem as with other approaches to iterative development
- Team members may be unsuited to the intense involvement that characterises agile methods

2 Kanban Boards

Used to track itemised work at any level:

- Arrangement of columns to track work progress
- Each column represents a step in the development process
- Each user story tracked across the columns
- Used to define next steps to be completed
- Easy to identify progress and bottlenecks

3 Agile

3.1 User stories

Definition: User stories

User stories are part of an agile approach that helps shift the focus from writing about requirements to talking about them. All agile user stories include a written sentence or two and, more importantly, a series of conversations about the desired functionality

- Captures the spirit
- Ignores details
- Make sense to customer
- Delivers value to customer
- End to end (full stack)
- Independent
- Testable
- Small (1-5 days) so easy to estimate

3.1.1 Behaviour Driven Development

Definition: Behaviour Driven Development

An agile process what supports and encourages collaborative development

Built on TDD and ATDD, plus:

- Where to start in the process
- What to test and what not to test
- How much to test in one go
- What to call the tests
- How to understand why a test fails