
Agile Methodologies

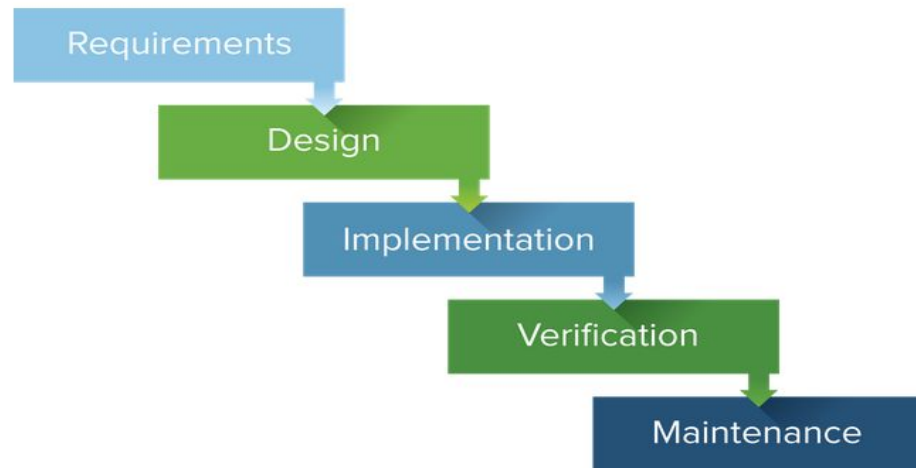
— Devharsh Trivedi —

Agile



- Continuous cycles
- Small, high-functioning, collaborative teams
- Multiple methodologies
- Flexible/continuous evolution
- Customer involvement

Waterfall



- Sequential/linear stages
- Upfront planning and in-depth documentation
- Contract negotiation
- Best for simple, unchanging projects
- Close project manager involvement

START

Initiate Project

Define
Requirements



Development
1

Development
2

Development
n

RELEASE

Accept

?

YES

Deploy
To
Customer

NO

Implement
Changes

Adjust, Track
Re-prioritize

Next Iteration
Into Development

**AGILE
METHODOLOGY**



AGILE & SCRUM TIP SHEET

4 AGILE VALUES

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 & 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.

12 AGILE PRINCIPLES



1. Our highest priority is to **SATISFY THE CUSTOMER** through early and continuous delivery of valuable solutions.



2. **WELCOME CHANGING REQUIREMENTS**, even late in development. Agile processes harness change for the customer's competitive advantage.



3. **DELIVER WORKING SOLUTIONS 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 SOLUTIONS** are 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.

3 SCRUM ROLES



PRODUCT OWNER
Key business decision-maker, prioritizer, and communicator



DEVELOPMENT TEAM
Cross-functional and self-organizing team of 5-9 people who do all the work to take backlog items to done



SCRUM MASTER
Coach, trainer, impediment remover, and servant leader for the team

5 SCRUM EVENTS

SPRINT PLANNING

PURPOSE - Understand WHAT the PO wants the Dev Team to work on and plan HOW to accomplish.



1. Backlog items should be small and well-understood by the Dev Team.



2. The Dev Team takes on a realistic amount of work based on capacity and past performance.



3. The Dev Team plans the work together with the goal of completing it together.

DAILY SCRUM

PURPOSE - For the Dev Team to synch their efforts, assess progress toward the sprint goal and plan their next day.



1. The format can vary but the focus is on hitting the team goals for the sprint.



2. The meeting should last less than 15 minutes. All dev team members attend.



3. Those outside the team may observe only.

BACKLOG REFINEMENT

PURPOSE - The team gets backlog items "ready" to increase likelihood of those items getting to done in a future sprint.



1. The Dev Team leads backlog refinement with input from the PO, SMEs and end users.



2. The Dev Team breaks backlog items down, adds details, and estimates backlog items.



3. Dev teams use good facilitation and definition of ready to limit refinement to no more than 10% of their capacity.

SPRINT REVIEW

PURPOSE - Demonstrate progress, showcase the team's results and get feedback on the product.



1. The Dev Team should show actual working results from the user's perspective. Don't show lines of code or PowerPoint.



2. Get organized, start on time and be succinct. Leave time for stakeholder discussion and feedback.



3. Expect feedback including new requests.

RETROSPECTIVE

PURPOSE - Allow the team to pause, reflect on their performance and identify ways to improve.



1. Retrospectives are owned by the Dev Team and the team decides who should attend.



2. Assume that everyone did the best they could under the circumstances.



3. Go deeper with root cause analysis. Select just one or two improvement actions each sprint.

Who is The ScrumMaster?

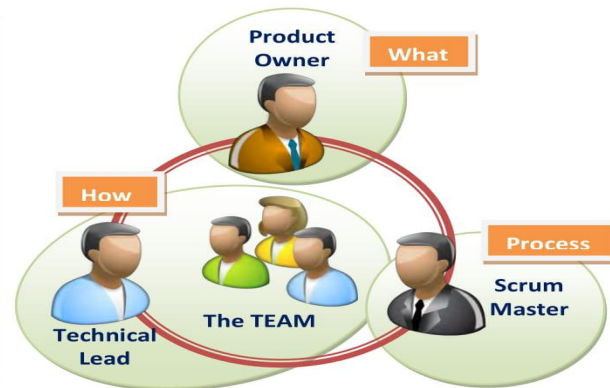
- Accountable for the **Process**
- Builds **empowered self organizing** teams
- Is a **Servant Leader**
- Removes **impediments**
- Creates **visibility, measurement, accountability** and fun
- Strong **facilitator**
- Encourages **collaboration** and open **communication**

Who is The Product Owner?

- One Person, accountable for **Backlog** and **Scope**
- **Prioritizes** the backlog
- Communicates the **Vision** of the product
- Defines **conditions of satisfaction**
- **Accepts/rejects** stories
- **Accountable** for project success
- **Accountable** for **ROI**
- Helps define **'Done'**
- Solicits **Customer** and **Stakeholders** feedback

Who is The Team?

- Cross-functional team accountable for **delivering business value** iteratively
- **Engaged, Motivated, Self Organizing, Collaborative**
- **Commits** to work
- **Inspects & Adapts**
- **Plans** and **re-plans**
- Delivers **'Done'** stories each iteration
- Uses **TDD, Automation** and **Best Practices**



Glossary

- **BACKLOG**: one list containing all stories
- **RELEASE PLAN**: rough schedule of iterations
- **RELEASE**: Moving 'Done' stories to production
- **BURN UP CHART**: demonstrates visually how many points the team got 'Done'
- **VELOCITY**: how many points the team got 'Done' in an iteration.
- **STORY**: a description of a small valuable customer requirement
- **FEATURE/THEME**: grouping of related stories
- **TASKBOARD**: where the team tracks their tasks visibly.
- **SCRUM MASTER**: owns the process, leads the team by empowering them, removes impediments, tracks progress.
- **TEAM**: cross-functional group working together to get a story 'Done'.
- **STORY POINTS**: a relative measure of complexity for a story.
- **IMPEDIMENT**: Anything stopping progress on a task.

The Iteration/Sprint Cycle



Story Format: 'As a <role> I want to <action> so that <value>

A Story Should Be:

Understandable

Independent

Negotiable

Valuable

Estimatable

Small

Testable

Story Points



Scrum Process

Enter your subhead line here

