AGILE

Waterfall; Government uses it; traditional approach to project management

WATERFALL --> Requirement--> Design-->Implementation-->Testing-->Deployment-->Maintenance

Step by step manufacturing models (airplane, cars)

Feasibility, Plan, Design, Build, Test, Production, Support

AGILE; **4 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 has better software quality, because we don't need to stop whole process

GUIDING PRINCIPLES

1. Customer Satisfaction
2. Welcome Changing Requirements
3. Working Software is Delivered Frequently (Weeks rather than months)
4. Close, Daily Cooperation between Business People and Developers
5. Project are built around motivated individuals, who should be trusted
6. Face-to Face Conversation is the best form of communication
7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity - The art of maximizing the amount of work not done - is essential
11. Best architectures, requirements and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

What is Agile?

Balance - Coordination - Speed - Reflexes - Strength - Endurance

Agile is a mindset (NOT bookish definition). Agile comes from TOYOTA

Agile **Methodologies** like; Kanban, *SCRUM*, XP, DSDM, Crystal, Cl, Lean (starting point), RUP

Agile **Component**: Framework

Framework: is a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful

User Story: is a tool used in Agile Software Development to capture a description of a software feature from an end-user perspective. It describes the type of user, what they want and why. It helps to create a simplified description of a requirement.

It comes from directly Business People.

As a, …. (who)

I want to … (what)

So I can … (why)

SCRUM

- is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software ***(2-4 weeks)***
- Business sets the priorities. Teams self organize to determine the best way to deliver the highest priority features.
- Every 2 weeks to 4 weeks anyone can see real working software and decide release it as is or continue

SCRUM FRAMEWORK

- **ROLES**;
    - **Product Owner**
        - Defines and Prioritize and Adjust features of the product (Functionality)
        - Decide on release date and consent
        - Responsible of profitability of the product
        - Accept or reject works results
        - Leading and motivating the team
        - Communicates with client, stakeholders and being a bridge between development team and client
    - **Scrum Master** (Are we following the rules - Coordinator - Football coach)
        - Responsible for enacting scrum Values and Practices
        - Represents management to the project
        - Removes Impediments (obstacles, blocker - reduce to progress - Example; I need a computer or app to do my job)
        - Close cooperation across all roles and functions
        - Shield the team from external interferences
        - Do everything possible to make sure development team performs in highest level
        - Here to serve team, he is not authority
        - Coordinates the collaboration (engagements) among the team member and develop the team mindset
    - **Team**

        ME:
        Sprint cycle is 2 weeks
        Release cycle is 3 months

        6 sprints in 1 release cycle

        - **SCRUM TEAM**
            - Team share the same Norms and Rules
            - Is Empowered
            - Autonomous
            - Self organizing
            - The skills within the scrum team are Balanced
            - A scrum Team is small (4-9 people) and has no sub-teams
            - Work full Time
            - People are collocated
            - Everyone works together, there is no QA or Dev Team

- **CEREMONIES**;
    1. **SPRINT PLANNING MEETING**

        1st part
        - Analyze, evaluate and select Product Backlog
        - Establish Goal and forecast work

        - Team needs to know how much **capacity** and commit some **points** (each points represent some **hours (individually)**
        - **Definition of done** is clarified by the team for each story
        - **Sprint backlog** is created at the end of the meeting (2 hours)
        - Points scale (1, 2, 3, 5, 8, 13, 21)

        2nd part
        - Figure out how to do work; decompose it into actionable work plan
--> Sprint Backlog

        Sprint is over WHEN the TIMEBOX (Iteration) expires.

2. **DAILY SCRUM MEETING (DAILY STAND-UP)**
   - Everyday but the first day of the sprint
   - Around 15 mins
   - Same Place and time with Full Team
   - Run by SCRUM MASTER
   - Provide Update on the sprint backlog work progress
   - Focus on 3 Main Questions
     - What I did yesterday
     - What I'll do today
     - What is in my way (any blocker?)
   - There can be HOT TOPIC at the end of the meeting

3. **SPRINT DEMO (REVIEW) MEETING**
   - Usually it happens at the end of the Sprint (like after the Every 2 sprint)
   - Usually everyone will join (+ customer, stakeholders …)
   - All new functionalities of the app will be shown
   - Usually run by the developer
   - Automation engineer can DEMO their automation script as well

4. **SPRINT RETRO (Retrospective) MEETING**
   - Usually happens after the Review (Demo) meeting
   - Usually everyone will join
   - Discuss the sprint;
     - What worked or went well?
     - What went wrong, caused problems, failed to work properly?
     - What should we do differently next spring to improve process?

NOT: SPRINT GROOMING MEETING

- Can happen multiple times in a sprint
- Usually PO and Scrum Team will join
- Requirements (written by BUSINESS PEOPLE) make up multiple epics --> Each epic can cover some features --> Each features have some user stories that are NOT prioritized
  - *Epic*; is the biggest user stories. It is almost everything about one project.
  - *Feature*; is multiple user stories that you can put together and it becomes stand alone component of one application.
- Story get prioritized for the up coming sprint (Scrum Master)
- Scrum Team decides which stories will come to the sprint
- NOTE: this meeting can sometimes be used to further understand the current sprint stories as well.

- ARTIFACTS;
  - **Product Backlog,**
  - **Sprint Backlog,**

```
O--> User story
OOOOOOO
OOOOOOO
OOOOOOO          OOOO
OOOOOOO    -->   OOOO   -->   SPRINT   -->    Shippable --> Happy Customers
OOOOOOO          OOOO        Daily Scrum       Product
Product Backlog    Sprint Backlog
```

  - **Burndown Charts**

    makes the work of the Team visible. It is a graphic representation of the rate at which work is completed and how much work remains to be done.

SHOW STOPPER (Blocker)

- When the environment is down, you are blocked
- When the environment is up, you continue to work
- When you can not login to system, you are blocked
- When you don’t have access to database, you are blocked
- When your test, test case, work has dependency to smo else's work, you might get blocked
- Test case dependency, work dependency (I am sick), Code dependency
- With less or no dependency, you are more productive
- What do you do when you have blocker issue? --> Go Scrum Master

DONE

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

BIGGEST CHALLENGE is

If developers give us the product very late, and then for instance we could not have enough time to finish the testing at the end of sprint, what is going to happen?
So it is the TIME issue.