Agile – It’s a Project Development & Delivery Methodology

SDLC – Software Development Life Cycle ( the different phases/stages of software development process)

WaterFall Model – Traditional SDLC Model (Planning, Designing, Coding, Testing, Releasing & Enhancing)

SDLC - SDLC is the series of steps that we go through when creating new products.

<https://agilemanifesto.org/>

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

Agile Documentation

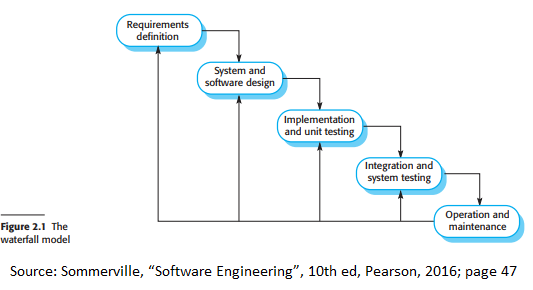
1. Product Backlog ( All the requirements/ user stories)
2. Sprint Backlog ( Shortlisted Requirements for a particular sprint)
3. BurnDown Chart (Graphical Representation of Effort utilisation

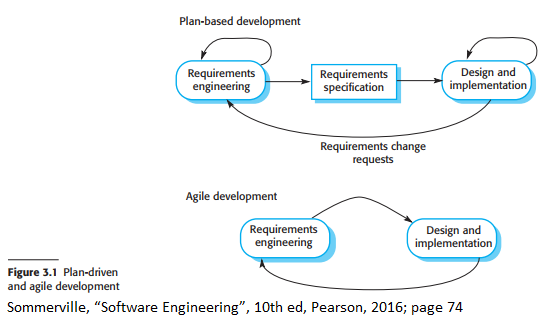
Agile Team consist of

1. Product Owner (Requirement) - Gives us the requirement
2. Scrum – Master (Manages all the scrum team members)
3. Scrum – Team ( Write, Test and deploy the code)

There won’t be any project managers, Quality control audit, tester in agile process.

Each Sprint = Planning + Designing + Coding+ Testing





User Story –

As a \_employee\_\_\_,

I like to \_login to banking application\_\_\_\_,

So that, \_I can transfer money\_\_\_\_

The complexity of a user story is measured in terms of story points (s,m,l,xl,xxl,xxxl – low, medium, high, 1, 2 3 )

Story Points (poker game)

Agile/Scrum Ceremonies == Meetings

1. Daily Stand up (Max 10 minutes – every one needs to stand and talk - what were you doing yesterday, what are you planning to do today, any roadblocks) [completed task/activity, planned task, roadblocks]
2. Sprint Planning – Beginning of Each sprint [ we will discuss, prioritise the requirements and select the user stories for the particular sprint]
3. Sprint Review & Retrospection – At the end of each sprint [ What went well, What we need to improve, what are the action plans] [Review – product demo, retro – process improvements]

P1 Front End choices

1. Angular Front End ( Angular application with components like login component, employee component, manager component) HTML, CSS, TS & Bootstrap as well as Material UI
2. JSP/Servlets [JSTL] [Simple backend & front end combined project]
3. HTML,CSS,JS – Fetch the backend API and update the UI

P1 Backend

1. Using Javalin/Spring based REST Api
2. Using SpringBoot & Data JPA REST Api
3. Maven Web App with Hibernate