

# Software Development Life Cycle and Methodologies



# Agenda

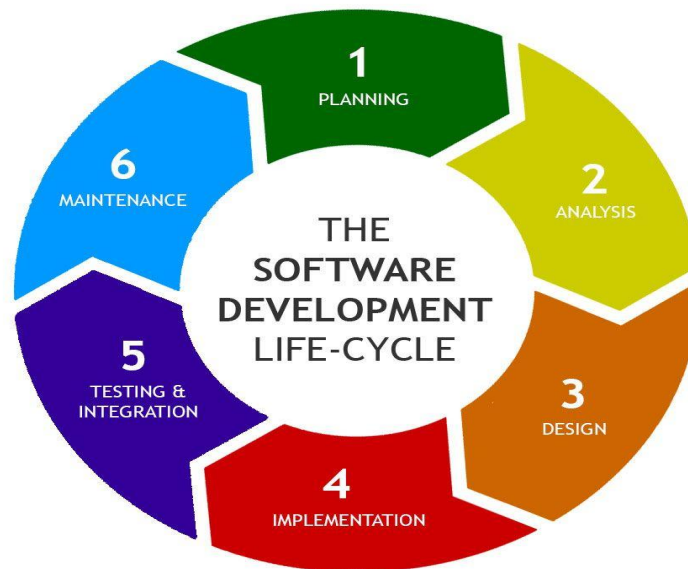
- Self-study, home work discussion. Questions.
- SDLC.
- Processes.
- What is Agile?
- Scrum, Waterfall, XP...
- Practice.

A hand-drawn illustration of the word "AGENDA" in a bold, black, sans-serif font. The word is written on a white background. A hand is visible at the bottom right, holding a black marker and drawing a horizontal line underneath the word. The hand is wearing a white sleeve. There are several faint, repeating watermarks of the word "AGENDA" scattered around the main text.

AGENDA

# What is SDLC?

The **Software Development Life Cycle (SDLC)** is a framework defining tasks performed at each step in the software development process. **SDLC** is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software.



# Processes.

Why do we care about it?



# For any company

- Customer satisfaction critical
- Project should end successfully all the time
- Project: complex NEW DIFFERENT software products
- Different size, different teams, different locations, distributed development

# SDLC gone wrong



How the customer explained it



How the project leader understood it



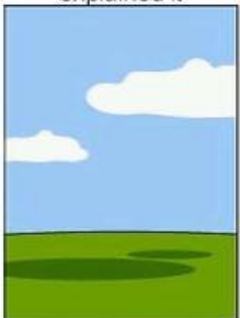
How the engineer designed it



How the programmer wrote it



How the sales executive described it



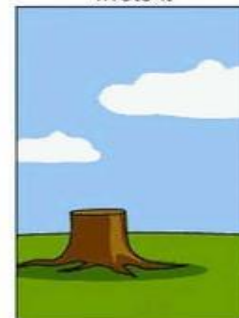
How the project was documented



What operations installed



How the customer was billed



How the helpdesk supported it

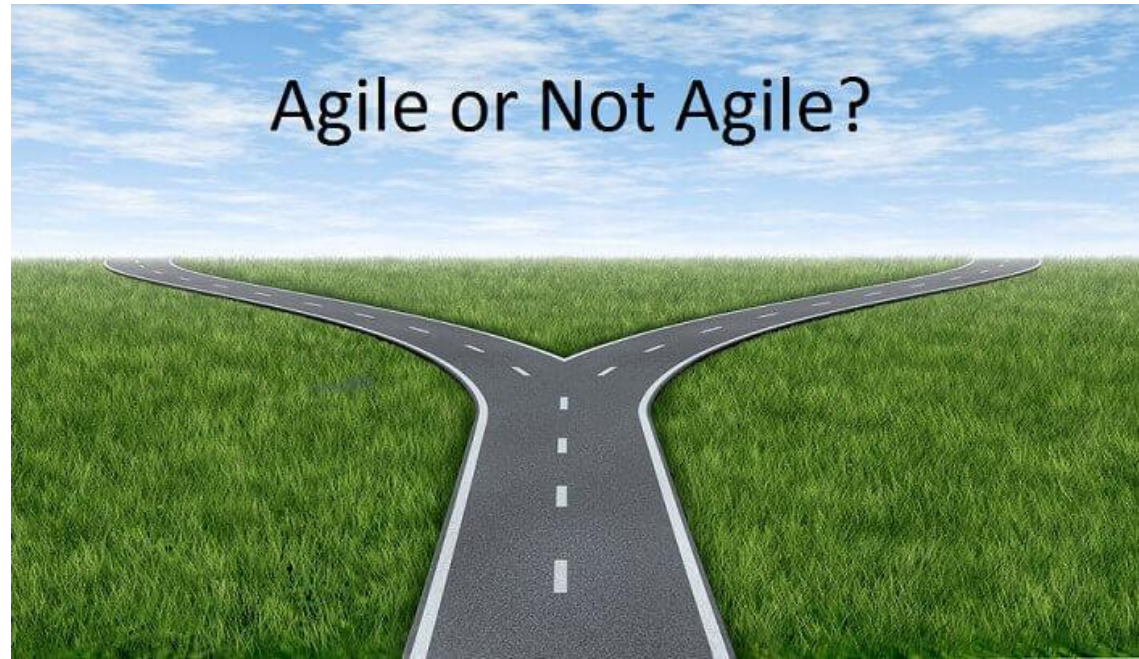


What the customer really needed

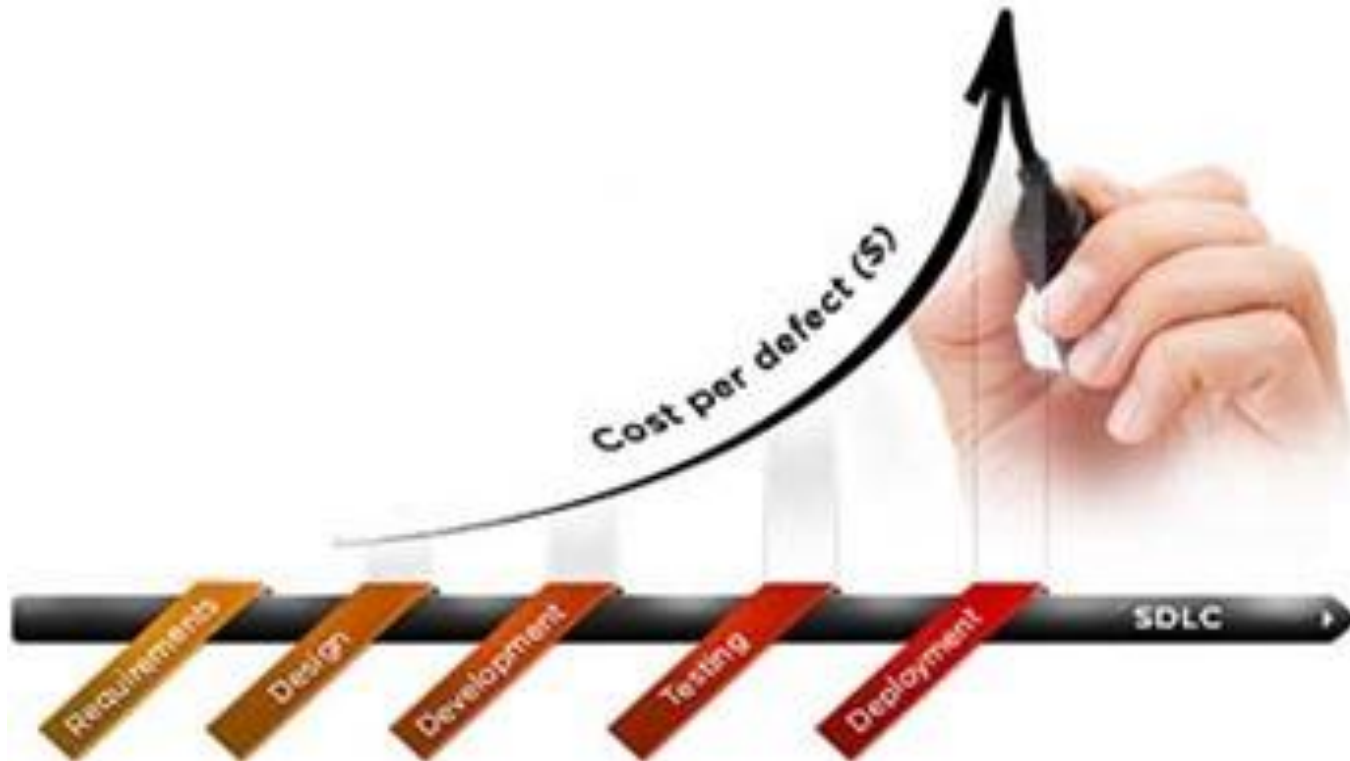


# Software Development Methodologies

- Waterfall
- Iterative
- RUP
- Agile
- XP
- ...



# Traditional Approach





# What is Agility?

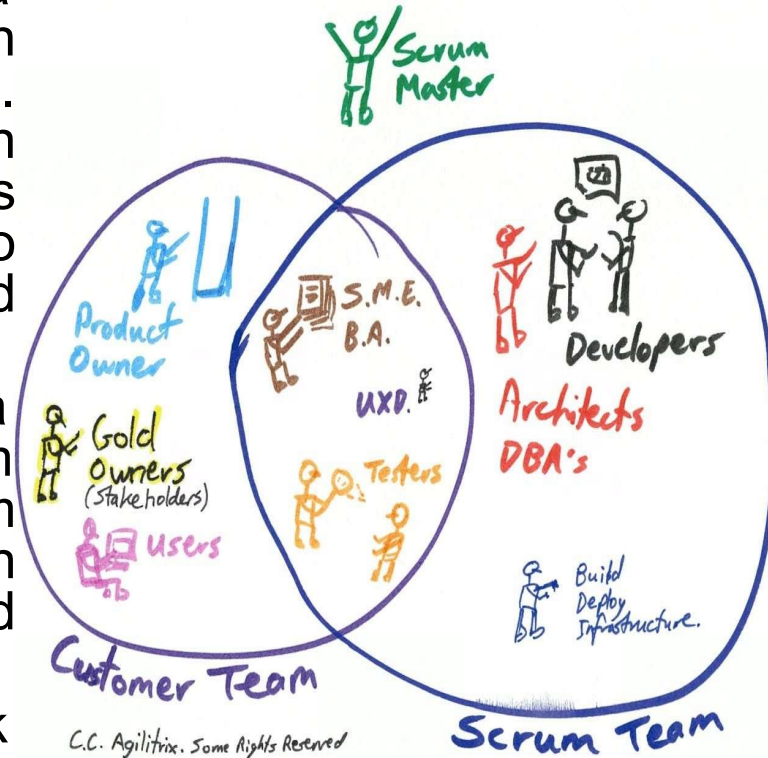
“Values, principles and practices that foster **team communication** and **feedbacks** to **regularly deliver** customer **value** through working software.”

# 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

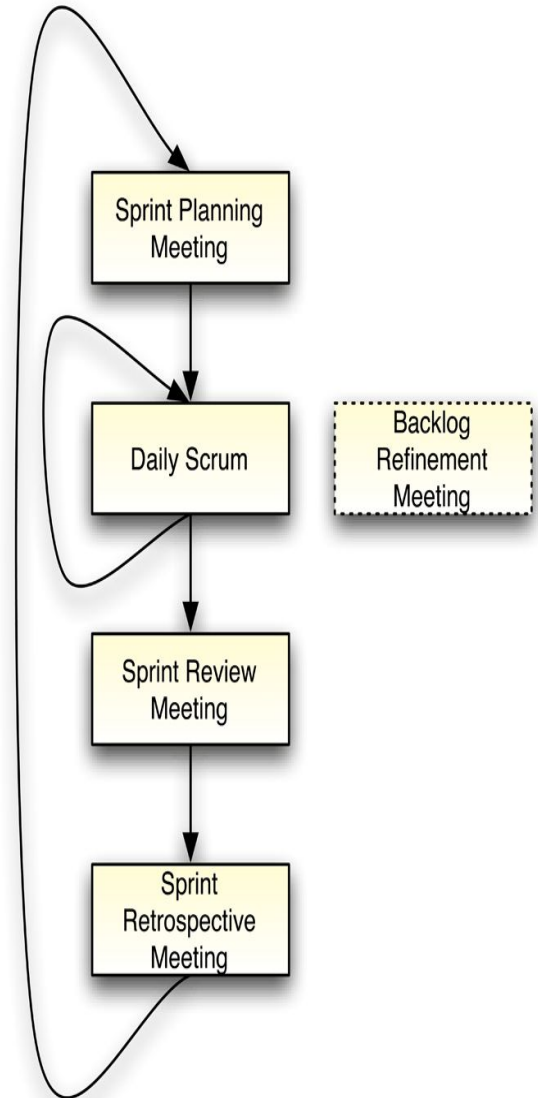
# Roles

- **Product owner** – The Product Owner is a member of the Scrum team charged with maximizing the value of the team's work. The product owner holds the product vision and works closely with stakeholders, such as end users, customers, and the business to cultivate and nurture a community around the product.
- **Scrum Master** – The Scrum Master is a servant leader, helping the rest of the Scrum team progress. He keeps the Scrum team productive and learning. He needs Coach the team, Keep the team moving forward and Help everyone understand Scrum.
- **Team** – The entire team (5-9 people), work together to achieve results. The development team does the actual work of delivering the product increment. The team is a cross-functional group of professionals who, among them, have all the necessary skills to deliver each increment of the product.



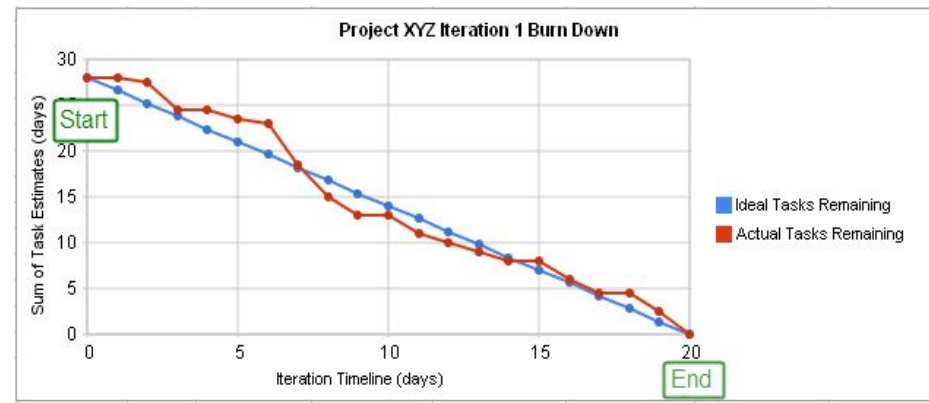
# Meetings

- **Sprint Planning** – the team meets the Product Owner to choose a set of items to deliver during the next sprint.
- **Daily Standup (Daily Scrum)** – the team meets each day to synchronize and identify impediments.
- **Sprint Review** – the team reviews and demonstrates to the Product Owner what is has completed during the Sprint.
- **Sprint Retrospective** – the team looks for ways to improve the processes.
- **Backlog Grooming** – the team meets with Product Owner to estimate product Backlog, refine the acceptance criteria for individual stories, and break larger stories into smaller ones.



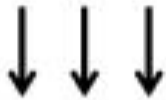
# Artifacts

- **Product increment** – an integrated, shippable product (build)
- **Product Backlog** – a prioritized list of desired project outcomes or features
- **Sprint Backlog** – a set of work from the Product Backlog that the team agrees to complete in a Sprint, broken down into tasks
- **Burn Down Chart** – view of the work remaining
- **Task Board** – active sprint board with the tasks in statuses like “To Do”, “In Progress”, “Done”



# Scrum Visualization

Inputs from End-Users, Customers, Teams and Other Stakeholders



Product Owner



Team

Features
1.
2.
3.
4.
5.
6.
7.
8.

Product Backlog

Plans how much work to commit for the Sprint

Sprint Planning Meeting

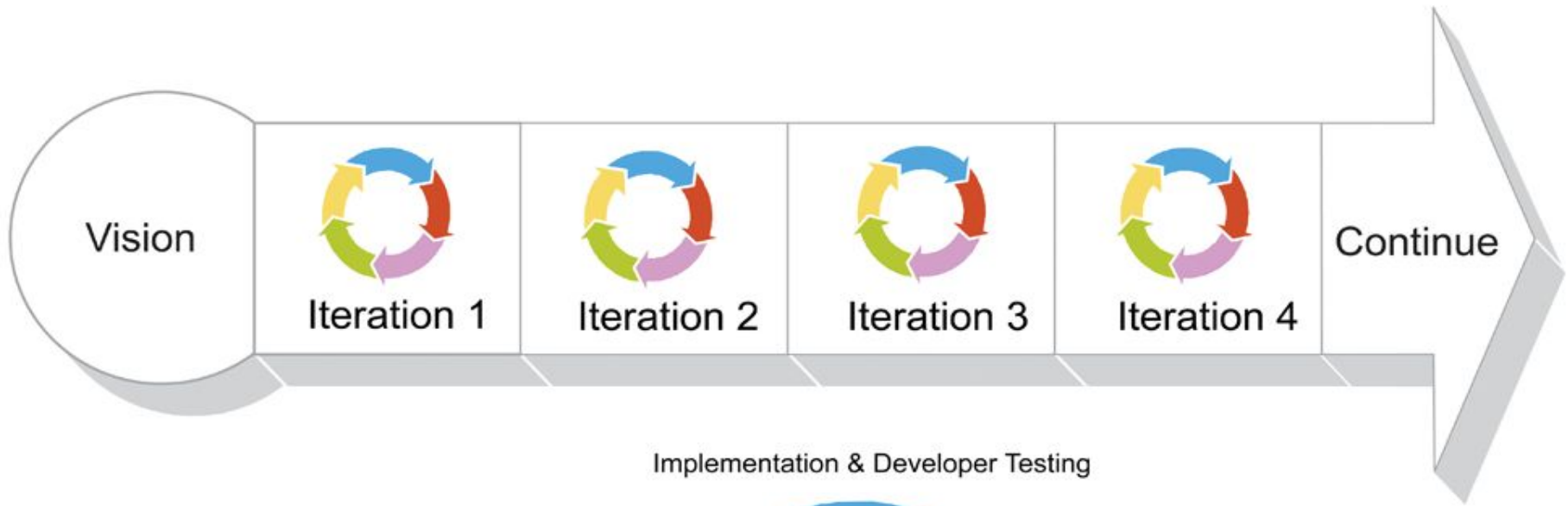
Tasks
1.
2.
3.
4.
5.

Sprint Backlog





# Result: Continuous improvements



Implementation & Developer Testing

## Iteration Detail



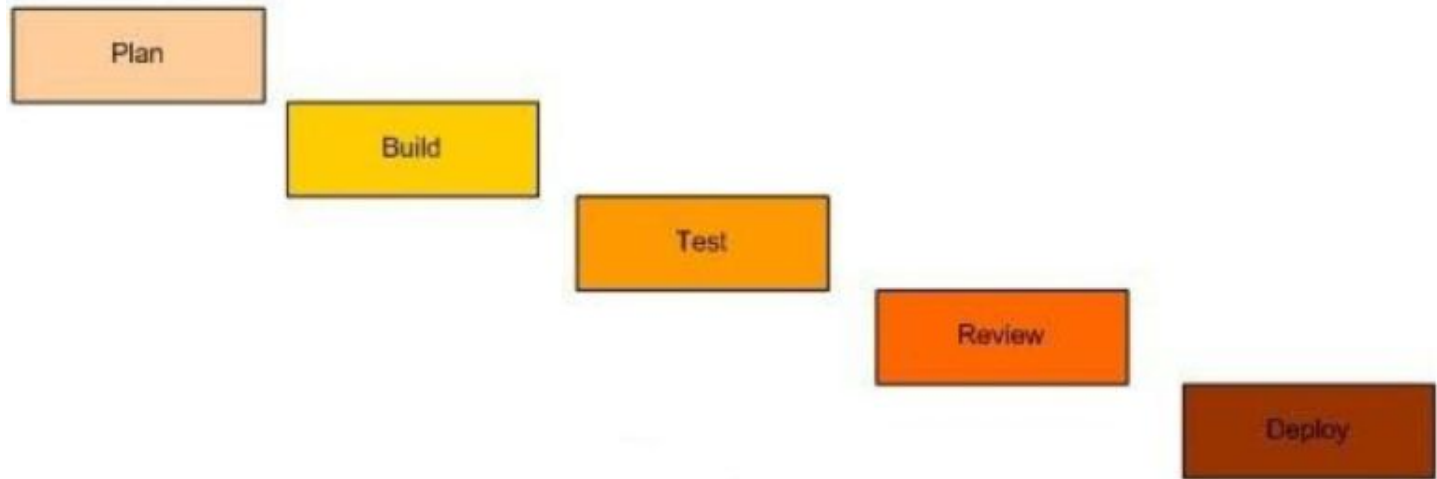
# Scrum vs Kanban

It is an **agile** process framework. Scrum and **Kanban** in software development are both specific shapings of an **agile** software methodology. While Scrum vs **Kanban** or **Kanban** vs Scrum is comparing two **agile** methodologies, Scrum vs **Agile** is comparing a concrete example with its fundamental principles.

Scrum	Kanban
Timeboxed iterations prescribed.	Timeboxed iterations optional. Can have separate cadences for planning, release, and process improvement. Can be event-driven instead of timeboxed.
Team commits to a specific amount of work for this iteration.	Commitment optional.
Uses <b>Velocity</b> as default metric for planning and process improvement.	Uses <b>Lead time</b> as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. <b>Specialist teams allowed.</b>
Items must be broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burndown chart prescribed	No particular type of diagram is prescribed

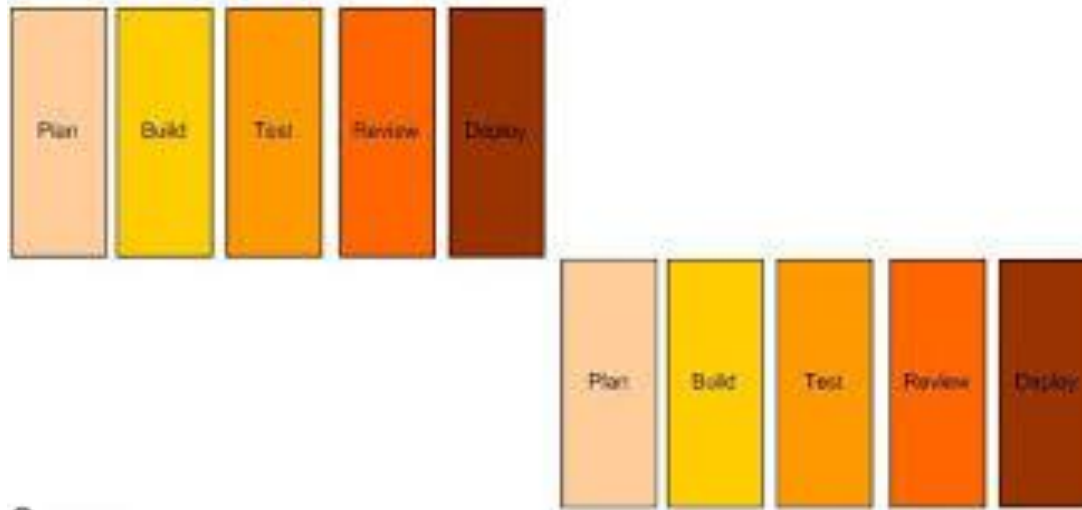
# Waterfall Development

**“Waterfall Development”** is another name for the more **traditional approach** to software development.



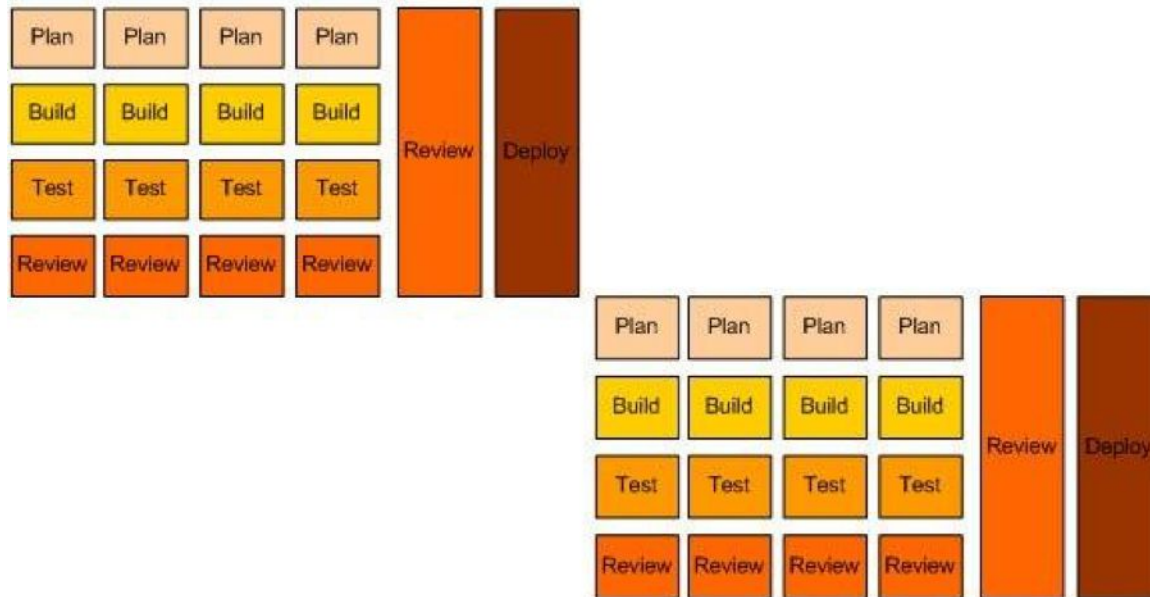
# Iterative Waterfall development

This approach carries **less risk than a traditional Waterfall approach** but is still far more risky and less efficient than a more Agile approaches.



# Scrum Development

This approach carries far less risk than Waterfall approaches. We focus on delivering fully-tested, independent, valuable, small features. We still plan our work in iterations and we will still release at the end of each iteration.

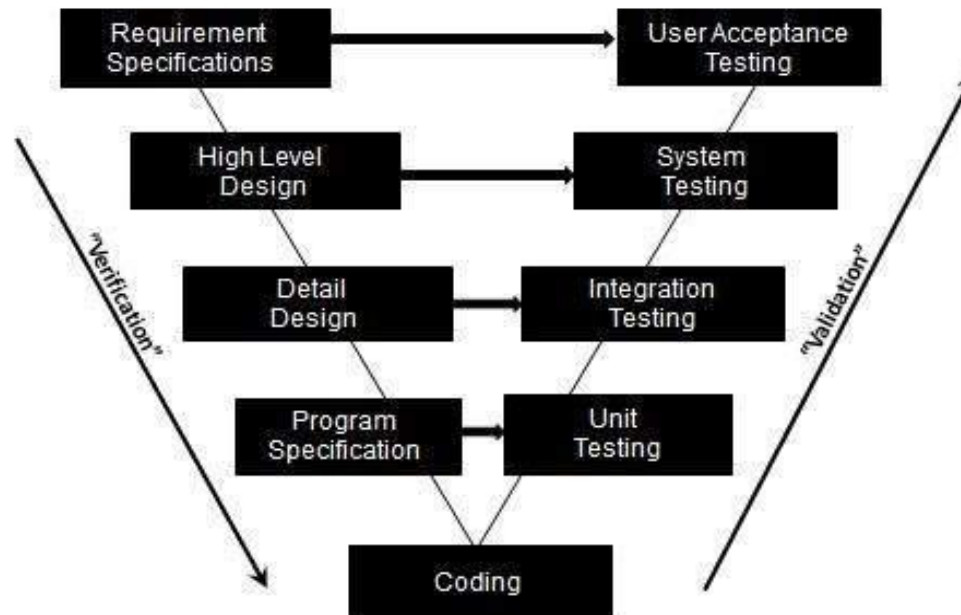






# V-model

The **V-model** is an SDLC **model** where execution of processes happens in a sequential manner in a **V-shape**. It is also known as **Verification and Validation model**. The **V-Model** is an extension of the waterfall **model** and is based on the association of a testing phase for each corresponding development stage.



# Scrum vs Waterfall in reality

Scrum takes its cue from RUGBY SCRUM where a team is aggressively trying to advance the ball and working together as one unit.

Scrum says we need to have a cross functional team and they have to be focused on advancing the common goal.



Waterfall method is the relay race approach where every individual is doing their part and then passing the baton to the next person.



# Practice: Scrum Penny Game



# Why Agile works?

- Less functional on iteration, less risks and wrong estimates.
- Fast rump-up (no need in deep review of docs)
- Often releases – REAL PROGRESS



**MR HAPPY CUSTOMER**

# Materials for self-study

- Book: Scrum and XP from the trenches (Link: <http://wwwis.win.tue.nl/2R690/doc/ScrumAndXpFromTheTrenchesonline07-31.pdf>)
- Article: <http://all4agile.com/en/differences-between-waterfall-iterative-waterfall-scrum-and-lean-software-development-in-pictures/>
- Video: <http://scrumtrainingseries.com/> (ALL)
- Scrum vs Kanban: <https://youtu.be/rlaz-l1Kf8w>
- Describe what methodology would you use to develop/test ... example