Overview of

**Agile Methodology**

**Prepared by: Rohith Chaitanya**

##### A [really] short history of

**Software development processes**

# Waterfall Development

**DESIGN**

**REQUIREMENTS**

**Waterfall Development** is another name for the more

**DEVELOPMENT**

**TESTING**

## traditional approach

**MAINTENANCE**

**to software development**

**Waterfall Development** (contd..)

You **complete one phase** (e.g. design) **before** moving on to the **next phase** (e.g. development)

You **rarely aim to re-visit a ‘phase’ once it’s completed**. That means, you **better get whatever**

**you’re doing right the first time!**

# But…



**Changes**

**REQUIREMENTNTS**

**DESIGN**

You don’t realize any value until



**DEVELOPMENT**

**Takes too long**

**Skipped**

**TESTING**

**MAINTENANCE**

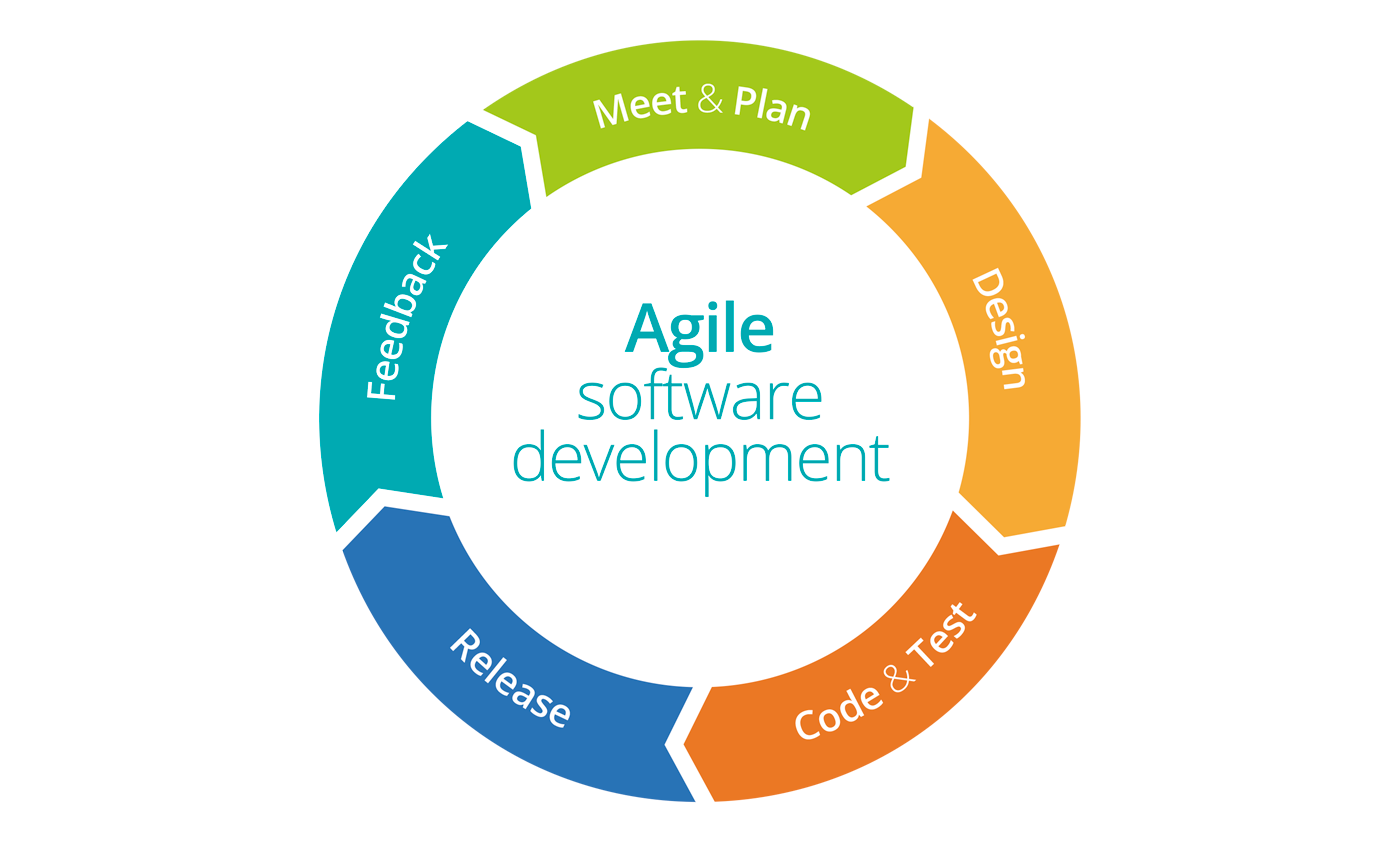
the end of the project

You leave the testing until the end You don’t seek approval from the

stakeholders until late in the day

This approach is **highly risky**, often more **costly** and

generally **less efficient** than **Agile**



### Individuals and interactions over

**processes and tools**

**Working software** over **comprehensive documentation**

**Customer collaboration** over



**contract negotiation**

**Responding to change** over

**following a plan**

follow

### light-weight agile process tool

**Scrum**

**Split your organization**

into small, cross-functional, self- organizing teams.

Product/ Project Owner

Scrum Master



Scrum Team

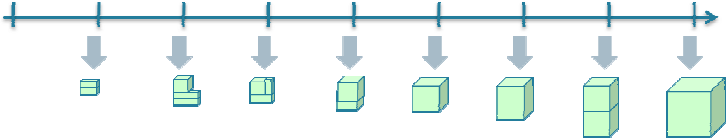
**Split your work** into a list of small, concrete deliverables.

Sort the list by priority and estimate the relative effort of each

item.

**Scrum**

**Split time** into short fixed-length iterations/ sprints (usually 2 – 4 weeks), with potentially shippable code demonstrated after each iteration.

January May

**Optimize the release plan** and update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration.

**Optimize the process** by having a retrospective after each iteration.

# Agile vs. Waterfall

**DEVELOPMENT**

**DESIGN**

**TESTING**

**REQUIREMENTS**

**MAINTENANCE**

**Iterative Scrum**



**Daily scrum meeting** to discuss **What did you do y’day**? **What will you do today**? **Any obstacles**?

**Scrum in a nutshell**

So instead of a **large group** spending **a long time** building a

**big thing**, we have a **small team** spending a **short time**

building a **small thing**.

**But integrating regularly to see the whole.**

