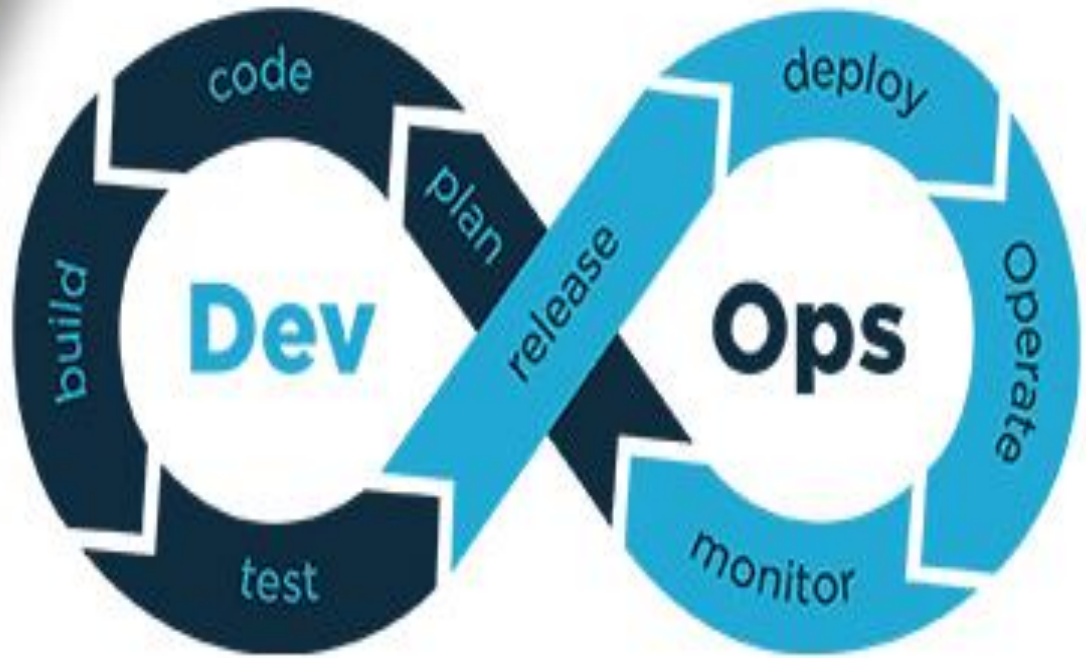
DevOps

PREPARED BY
ARAVIND KUMAR G.K
Senior DevOps Engineer



Agenda

- **What Is DevOps**
- **History of DevOps**
- **DevOps definition**
- **DevOps Main Objectives**
- **House Of DevOps**
- **DevOps and Software Development Life Cycle**
  - Waterfall Model
  - Agile Model

What is DevOps?

- DevOps is a culture which promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way.
- The word 'DevOps' is a combination of two words 'development' and 'operations.‘
- DevOps helps to increases an organization's speed to deliver applications and services. It allows organizations to serve their customers better and compete more strongly in the market.
- In simple words, DevOps can be defined as an alignment of development and IT operations with better communication and collaboration.

DevOps Myths

Popular DevOps Myths

- DevOps is about collaboration between Development & Operations teams
- DevOps is all about using Tools & Automation
- DevOps is a Skill
- Special DevOps teams are needed to practice DevOps
- DevOps disrupts existing processes like ITIL
- Anybody can do 10 deploys a day with DevOps
- DevOps Is for cloud or web shops only
- DevOps is a silver bullet
- Traditional projects cannot adopt DevOps

History of DevOps

DevOps Timeline

2007

Patrick Debois start's assessing IT Value Chain

2008

Agile System Administrators Group is launched on Google

Google

Agile2008 Conference

Andrew Clay Shafer and Patrick Debois meet at Agile Conference 2008

2009

Inaugural "DevOps Days" are held in Ghent, Belgium

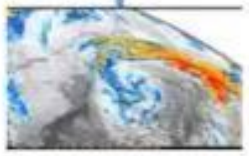
CHEF

Open Source toolsets rip up the legacy playbook.

Velocity CONFERENCE

John Allspaw and Paul Hammond present "10 Deploys per day" at Velocity

2010



The "Perfect Storm" of adjacent methodologies occurs

2011

Industry leading software vendors increase market presence with "Enterprise" class DevOps tools

IBM

hp

ca technologies

Industry leading software vendors increase market presence with "Enterprise" class DevOps tools

2012

2013

Gene Kim releases "The Phoenix Project"

DevOps begins to provide positive impact to "Enterprise" IT

2014

devX is born and Xceed launches the "12 days of DevOps"

2015

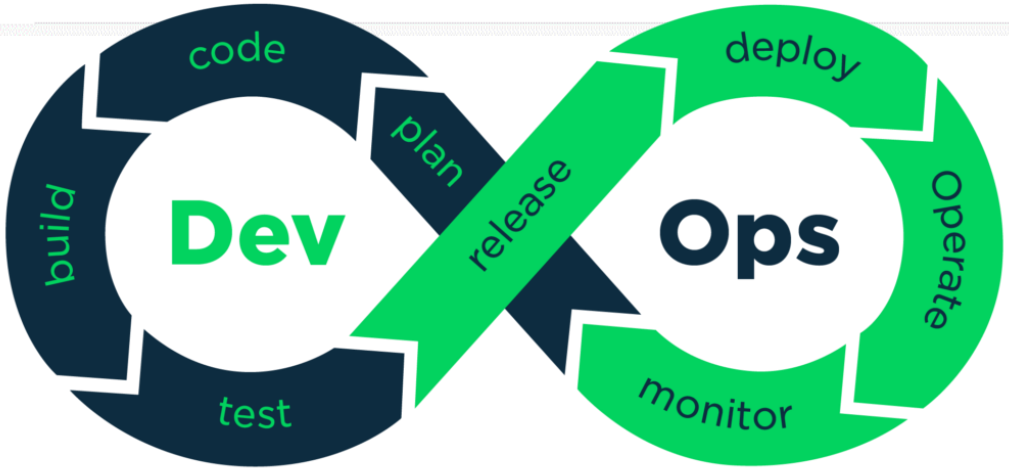
What does the future of DevOps hold for your organisation?

Definition of DevOps

DevOps

DevOps (a clipped compound of "development" and "operations") is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.

DevOps - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**DevOps**

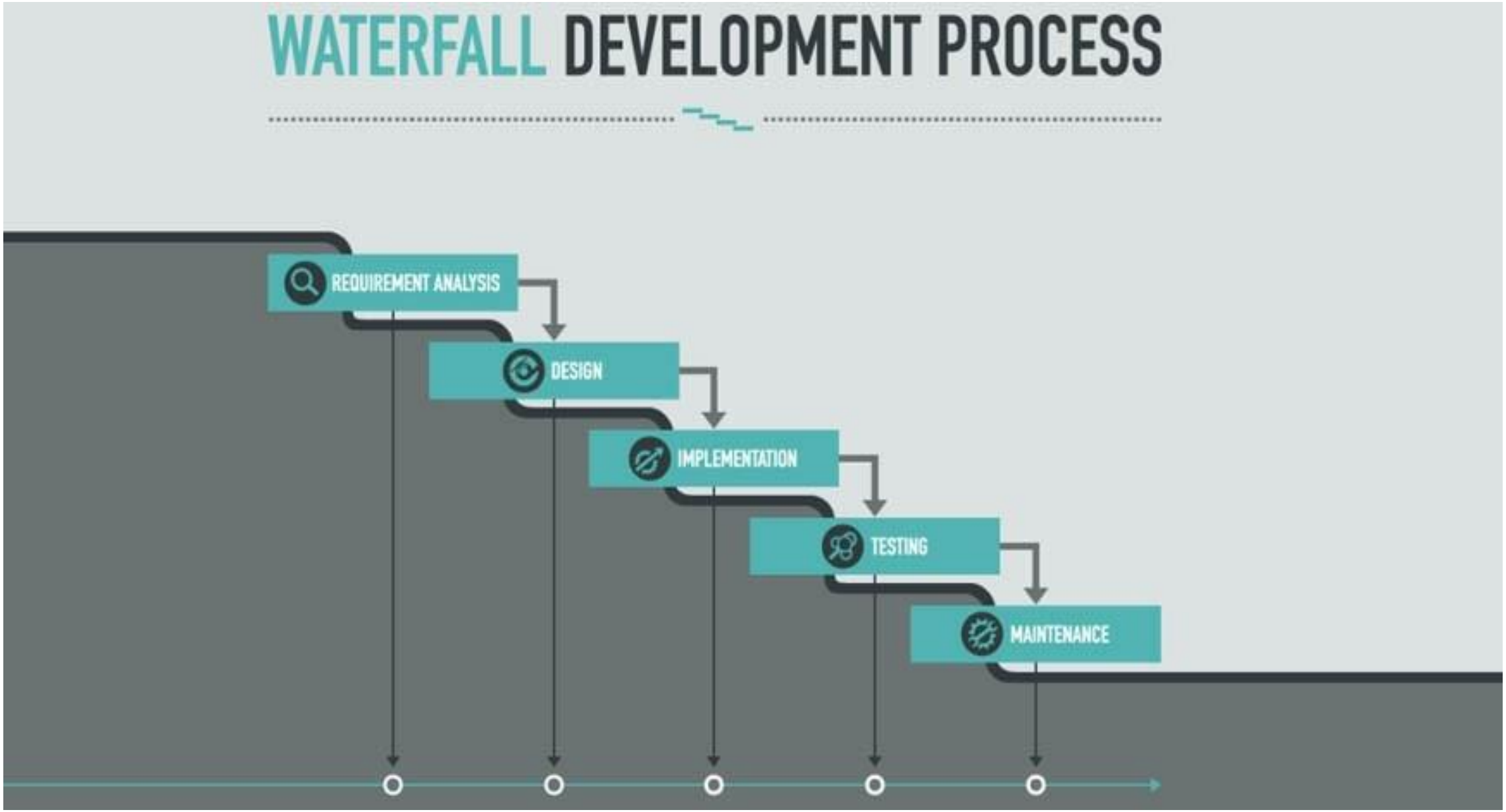


DevOps: Definitions

- "DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support." – http://theagileadmin.com/what-is-devops/
- "DevOps is the blending of tasks performed by a company's application development and systems operations teams." – http://searchcloudcomputing.techtarget.com/definition/DevOp
- "DevOps (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between Development and IT Operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between the two business units." – http://www.webopedia.com/TERM/D/devops_development_operations.html

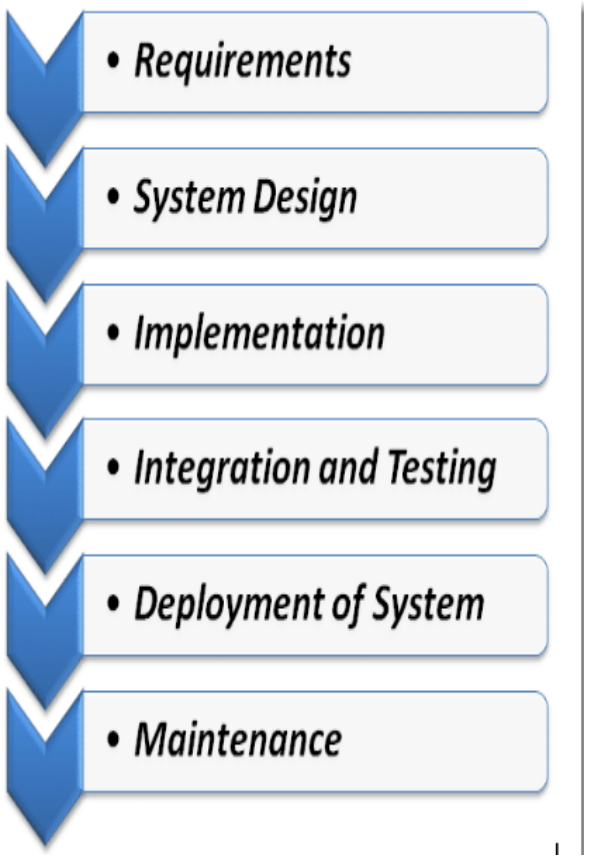
DevOps Main Objectives

- Increases the frequency and quality of deployments.
- Improves innovation and risk-taking by making it safer to experiment.
- Realizes faster time to market.
- Improves solution quality and shortens the lead time for fixes
- Reduces the severity and frequency of release failures.
- Improves the Mean Time to Recovery (MTTR)



Waterfall Approach to Development

- In *"**The Waterfall**"* approach, the whole process of *software development* is divided into separate phases.
- The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete.
- The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of ***Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.***
- It is also referred to as a ***Linear-Sequential Life Cycle Model***.



Sequential Phases in Waterfall Model

- **Requirements:** The first phase involves understanding what need to be design and what is its function, purpose etc. Here, the specifications of the input and output or the final product are studied and marked.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

Sequential Phases in Waterfall Model -Cont

- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.
- **Deployment of System:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Client is provided with regular maintenance and support for the developed software.

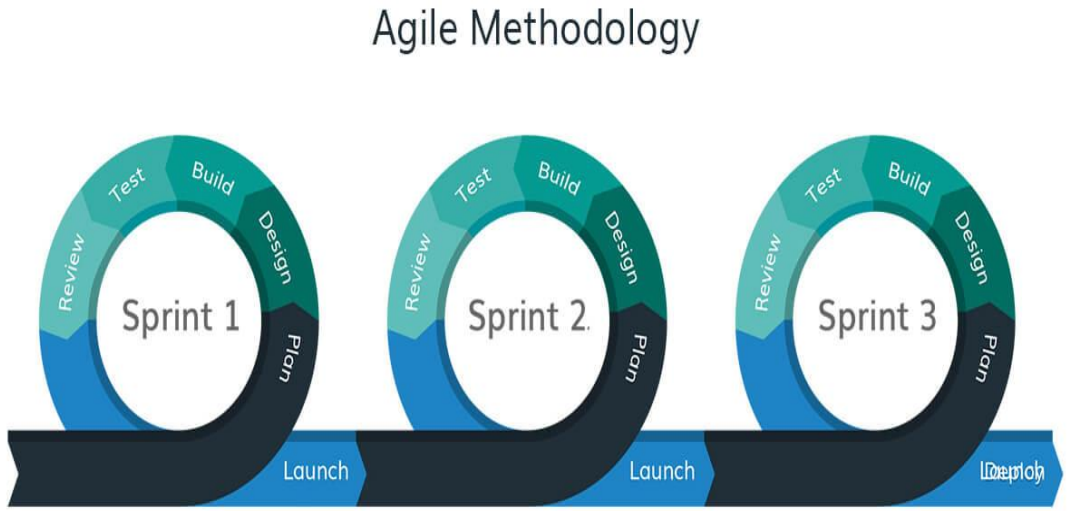
Water Flow Model Advantages and Disadvantages

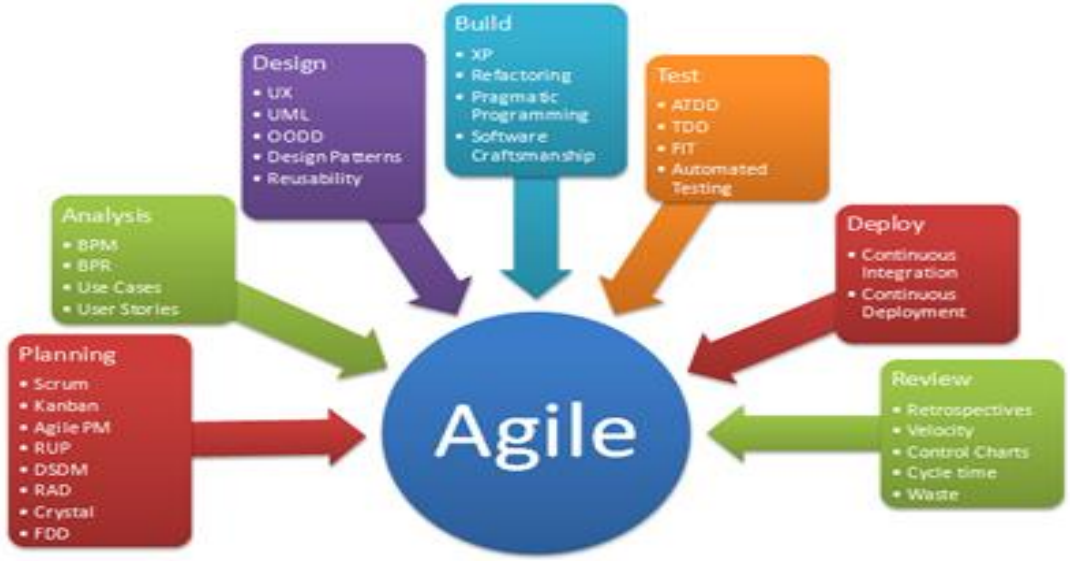
- **Advantages**
- The advantage of waterfall development is that it allows for departmentalization and control.
- The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model, phases are processed and completed one at a time and they do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

- **Disadvantages**
- It is difficult to estimate time and cost for each phase of the development process.
- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- Not a good model for complex and object-oriented projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

Agile Process

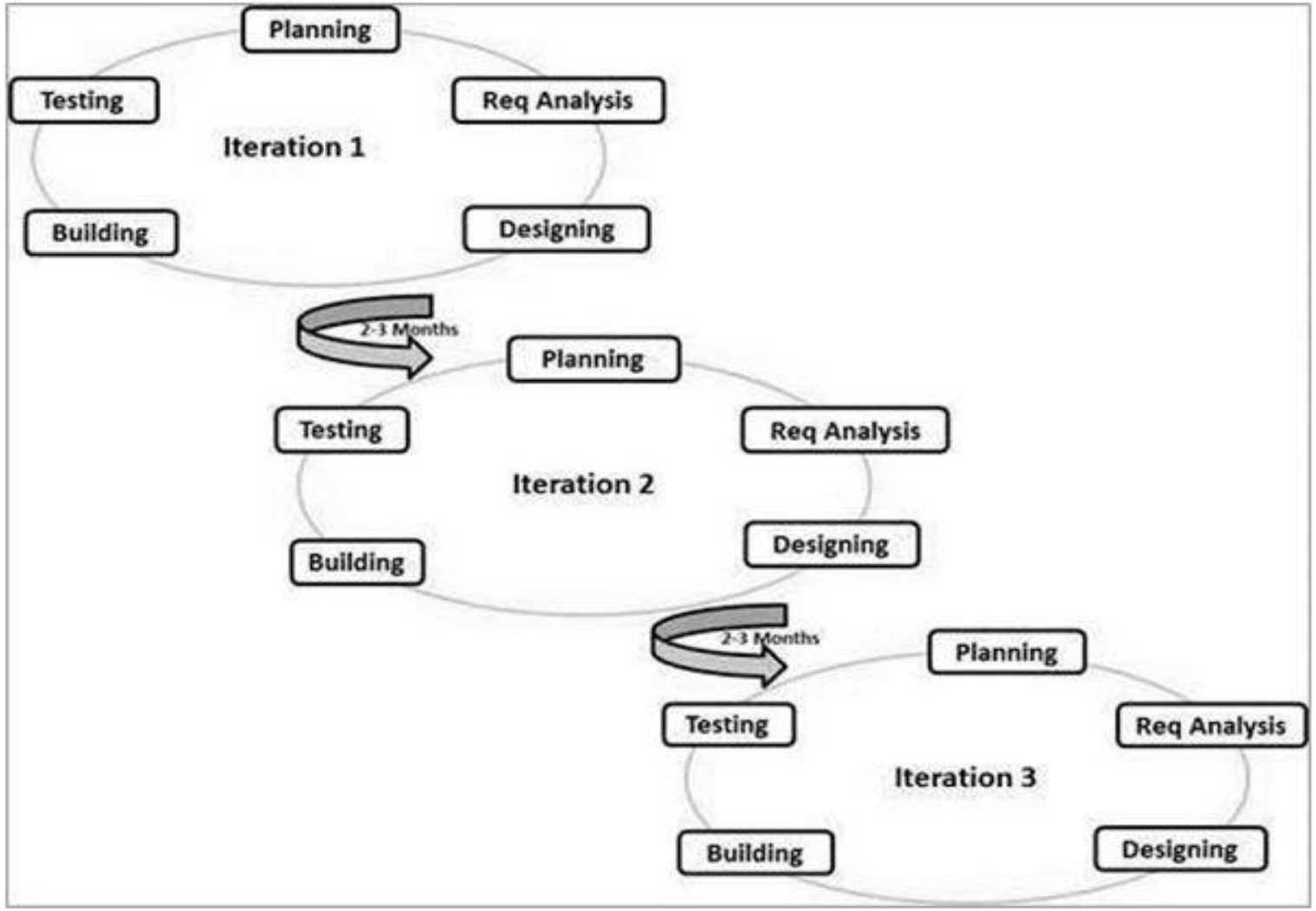
- Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product
- Agile Methods break the product into small incremental builds.
- These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.





Every iteration involves cross functional teams like

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.



Agile software development sprint planning

- Within the agile , work is divided into sprints. A sprint typically lasts two weeks, or 10 business days.
- **Plan.** The sprint begins with a sprint planning meeting, where team members come together to lay out components for the upcoming round of work. The product manager prioritizes work from a backlog of tasks to assign the team.
- **Develop.** Design and develop the product in accordance with the approved guidelines.
- **Test/QA.** Complete thorough testing and documentation of results before delivery.
- **Deliver.** Present the working product or software to stakeholders and customers.
- **Assess.** Solicit feedback from the customer and stakeholders and gather information to incorporate into the next sprint.

Advantages and disadvantages of Agile

Advantages

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Gives flexibility to developers.

Disadvantages

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.