What is Software Development?

What is Software Development?

Software Development is the process of transforming customer requirements into a complete software product.

Software Development Life Cycle

In broader terms, software development involves the following stages:

- Requirements
- Design
- Implementation
- Verification
- Maintenance

Software Development Life Cycle

- **Requirements**
- Design
- Implementation
- Verification
- Maintenance

This is the most important phase in the software development lifecycle. In this stage, the requirements are gathered from the customers and the requirements are then analysed to ensure their feasibility.

Software Development Life Cycle

- Requirements
- **Design**
- Implementation
- Verification
- Maintenance

Once the requirements are received, the architect transforms these requirements into technical specifications and plan the software components which have to be designed

Software Development Life Cycle

- Requirements
- Design
- **Implementation**
- Verification
- Maintenance

The specifications are then passed on to the developers which create the application based on these specifications

Software Development Life Cycle

- Requirements
- Design
- Implementation
- **Verification**
- Maintenance

Once the development work is done on the application. It is verified by a group of testers to map the application's functionalities with the specification given by customers

Software Development Life Cycle

- Requirements
- Design
- Implementation
- Verification
- **Maintenance**

Once the code is verified, it is pushed to production. Post this, the application is updated with any future enhancements or optimizations, if and when required.

SDLC Models

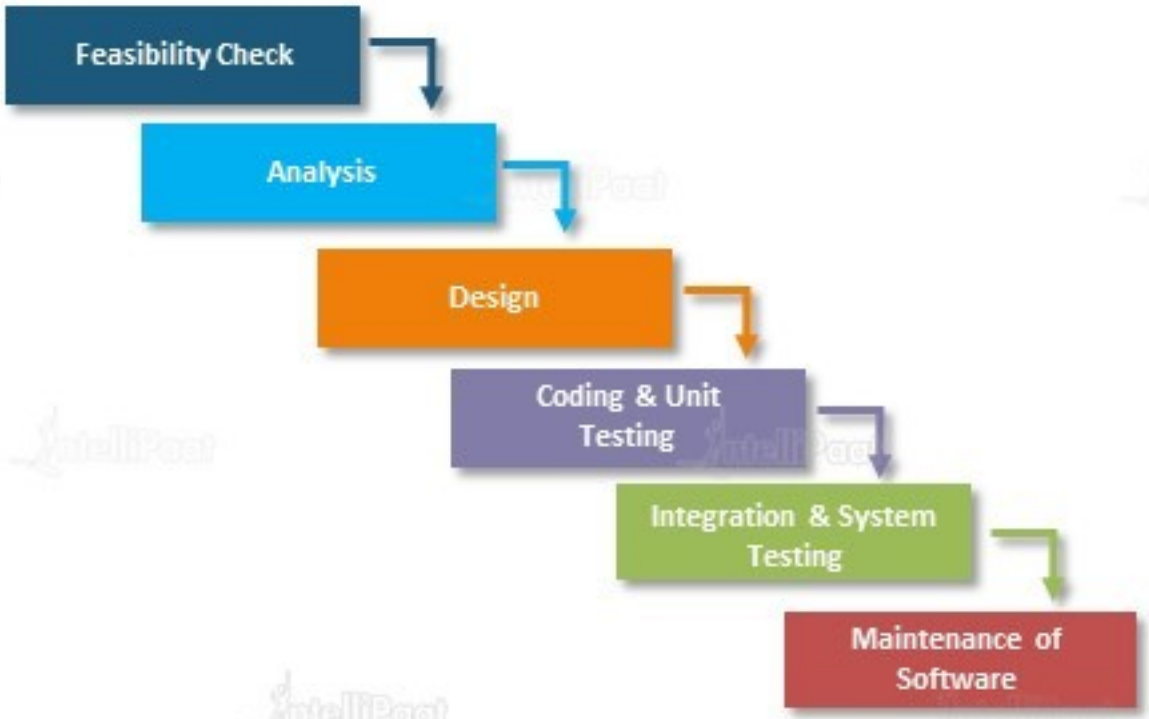
Since the time software development started, various software development models have been curated which implement SDLC. Each of these models solve problems that existed before these models were invented.

Traditionally, there have been 3 major software development models that most companies follow:

- Waterfall Model
- Agile Model
- Lean Model

Waterfall Model

Waterfall Model



- Waterfall Model was among the first development models which followed SDLC
- The Waterfall model follows a linear sequential model of development i.e until the first stage is not finished, the next stage will not start

Advantages of Waterfall Model

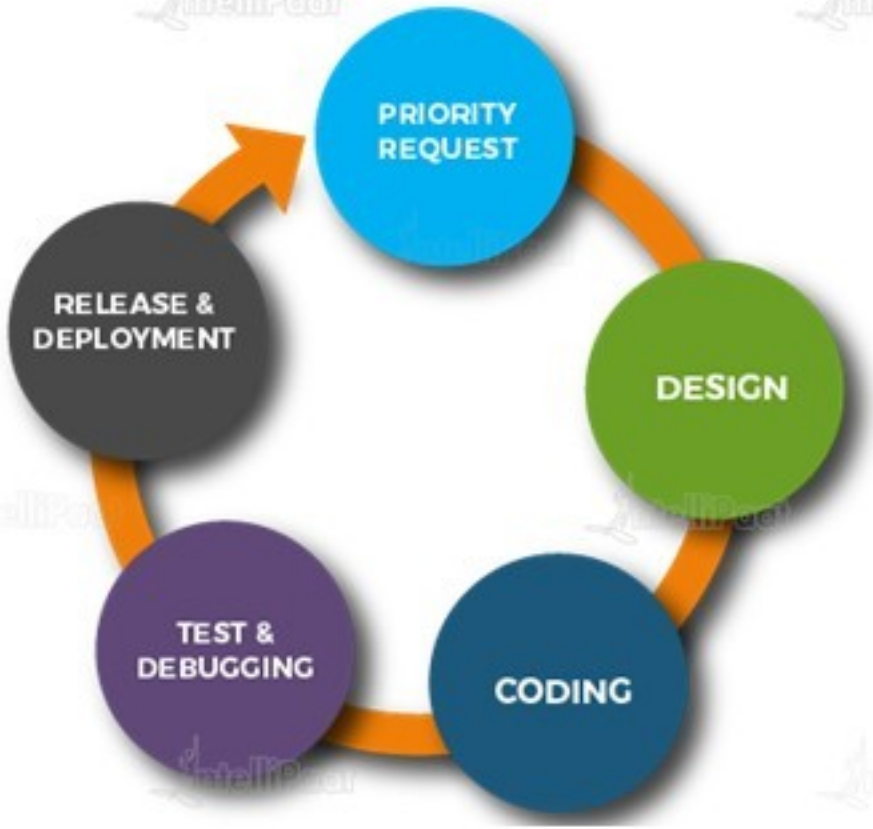
- Clear Objectives
- Specific Deadlines
- No ambiguous requirements
- Well understood milestones
- Process and results are well documented

Disadvantages of Waterfall Model

- Working Product is not available until the later stage in lifecycle
- Poor model for large and complex projects
- Cannot accommodate changing requirements
- High risk and uncertainty

Agile Model

Agile Model



- To overcome the challenges faced in the Waterfall Model, we came up with the Agile Methodology
- Agile Method believes in creating shorter development lifecycles
- Shorter Development Lifecycles are achieved by not releasing all the features at once by following an incremental model of development

Advantages of Agile Model

- Customer Satisfaction is high
- Less Planning Required
- Requirements can be dynamic in nature
- Functionality can be created and tested quickly

Disadvantages of Agile Model

- Not suitable for handling complex dependencies in projects
- Knowledge transfer to colleagues can be difficult since there is little documentation
- Success of the project depends heavily on customer interaction

Lean Model

Lean Model

7 Principles of Lean Methodology

- Eliminate Waste
- Amplify Learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build Integrity
- See the whole

- Lean development is a philosophy of increasing quality in software delivery by making use of agile methods
- Ignore the clutter for later and focus on what is required now
- Lean Methodology has it's primary focus on two things – Respect for frontline workers and Continuous Improvement

Advantages of Lean Model

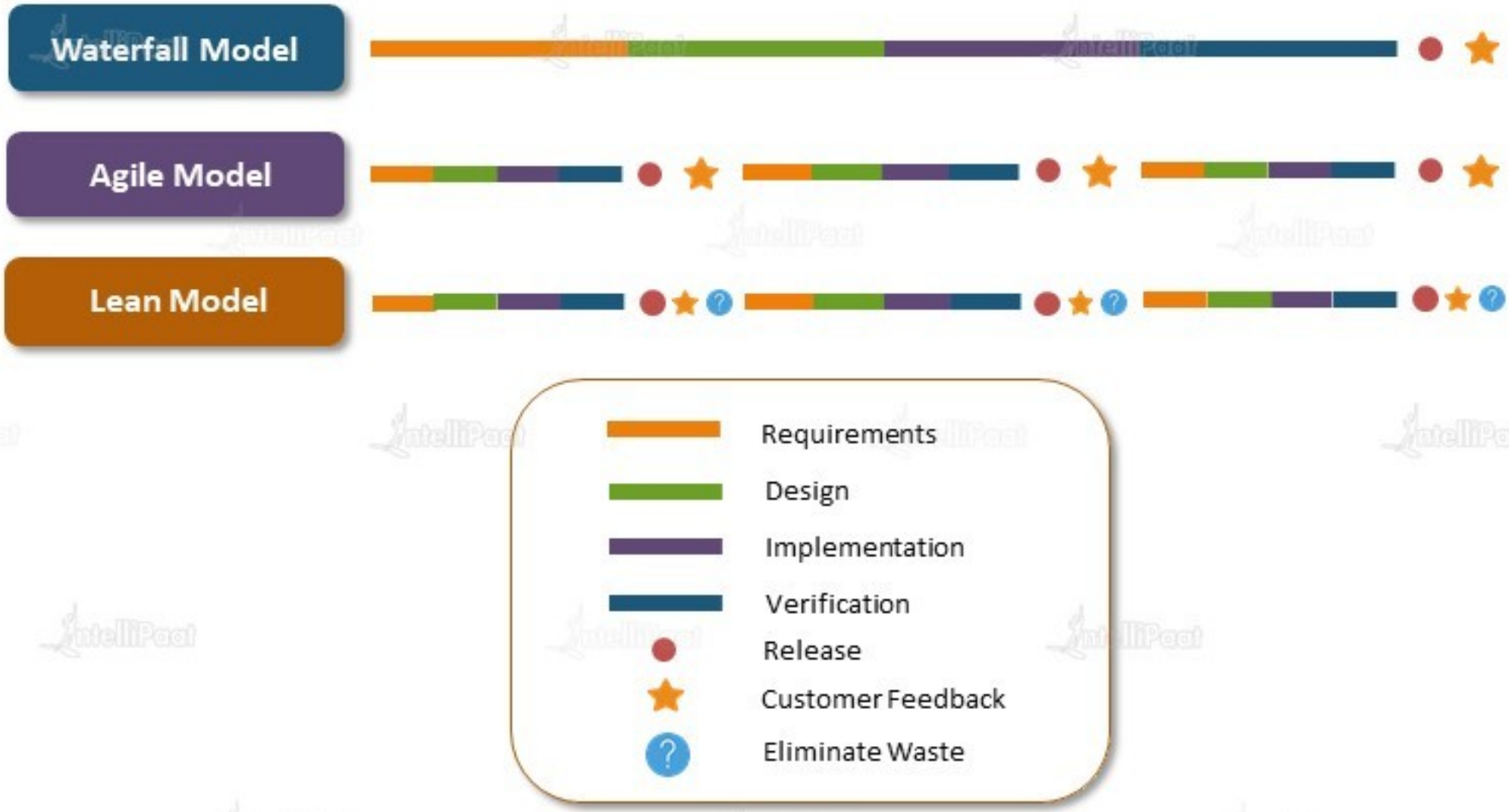
- Carries the same advantages as Agile Methodology
- Creates a positive working environment
- Customer Feedback is given the utmost importance
- Limiting Wastes saves time and money

Disadvantages of Lean Model

- Largely dependent on the skill set of the team, therefore requires a strong team
- No room for error, a missed delivery can be bad for business
- Success of the project depends heavily on customer interaction

Waterfall vs Agile vs Lean

Waterfall vs Agile vs Lean



Summarizing

Problem with Waterfall Model was, the development lifecycle took a lot of time to complete. Therefore, by the time finished product was delivered, the customer requirements were no longer the same.

Customers

Software Company

Summarizing

This problem was fixed by Lean and Agile methodologies. These methodologies strictly focussed on customer feedback and improving the software quality that too in a shorter development lifecycle

Customers

Software Company

Summarizing

This problem was fixed by Lean and Agile methodologies. These methodologies strictly focussed on customer feedback and improving the software quality that too in a shorter development lifecycle

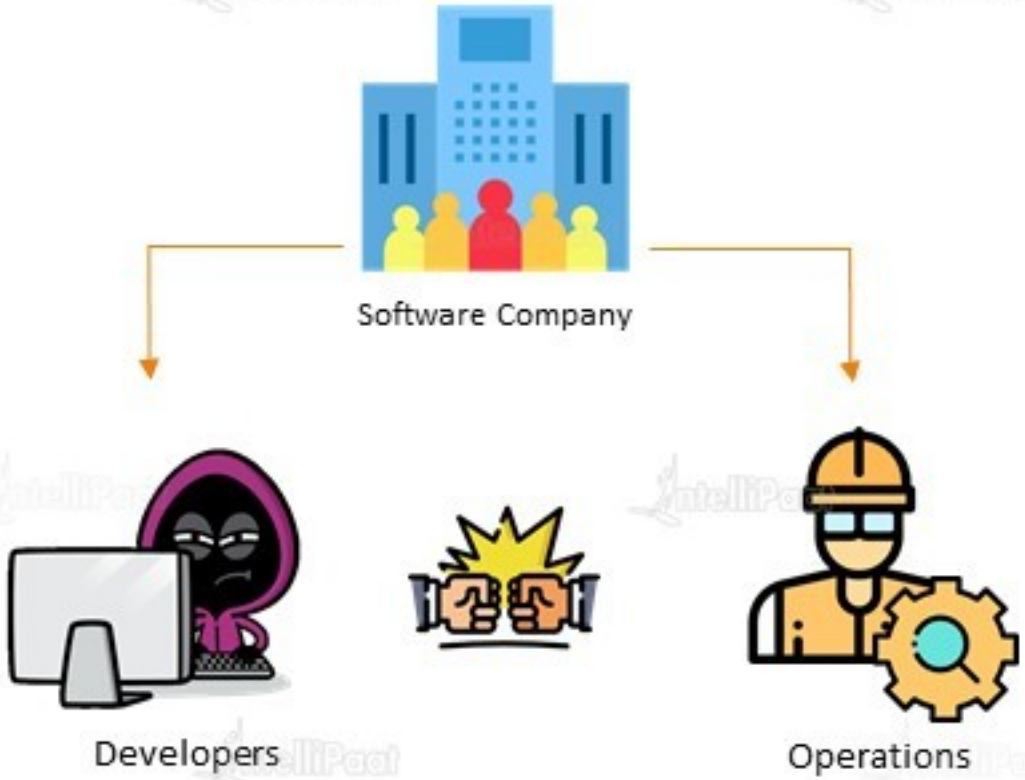
Customers

Software Company

Why do we need DevOps?

Why DevOps?

Why DevOps?



Although, the software quality was improved. We still had a lack of efficiency among the development team. A typical software development team consists of Developers and Operations employees. Let us understand their job roles

Why DevOps?

A developer's job is to develop applications and pass his code to the operations team

Developer

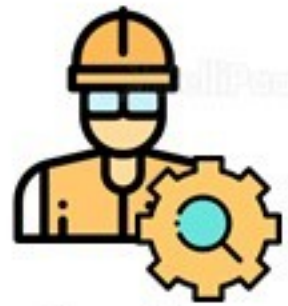
The operations team job is to test the code, and provide feedback to developers in case of bugs. If all goes well, the operations team uploads the code to the build servers

Operations

Why DevOps?

Developer

The developer used to run the code on his system, and then forward it to operations team.



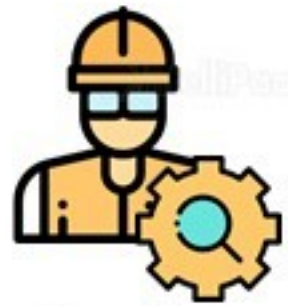
Operations

The operations when tried to run the code on their system, it did not run!

Why DevOps?

Developer

But, the code runs fine on the developer's system and hence he says "It is not my fault!"



Operations

The operations then marked this code as faulty, and used to forward this feedback to the developer

Why DevOps?

Developer

Operations

This led to a lot of back and forth between the developer and the operations team, hence impacted efficiency.

Why DevOps?

Developer

Operations

This problem was solved using Devops!

Traditional IT vs DevOps

Traditional IT	Devops
Less Productive	More Productive
Skill Centric Team	Team is divided into specialized silos
More Time invested in planning	Smaller and Frequent releases lead to easy scheduling and less time in planning
Difficult to achieve target or goal	Frequent releases, with continuous feedback makes achieving targets easy

Quiz

Quiz

1. Which of these Software Development Methodologies are not suitable for large and complex projects?

A. Waterfall Model

B. Devops

C. Agile Methodology

D. None of these

Quiz

1. Which of these Software Development Methodologies are not suitable for large and complex projects?

A. Waterfall Model

B. Devops

C. Agile Methodology

D. None of these

Quiz

2. Devops Methodology was focused on solving the problems between the customers and the software company.

A. True

B. False

Quiz

2. Devops Methodology was focused on solving the problems between the customers and the software company.

A. True

B. False

Quiz

3. Which of these principles are NOT included in Agile Methodologies?

A. Frequent Release Cycles

B. Focus on Customer Feedback

C. Eliminating Waste

D. None of these

Quiz

3. Which of these principles are NOT included in Agile Methodologies?

A. Frequent Release Cycles

B. Focus on Customer Feedback

C. Eliminating Waste

D. None of these

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24X7 Chat with our Course Advisor