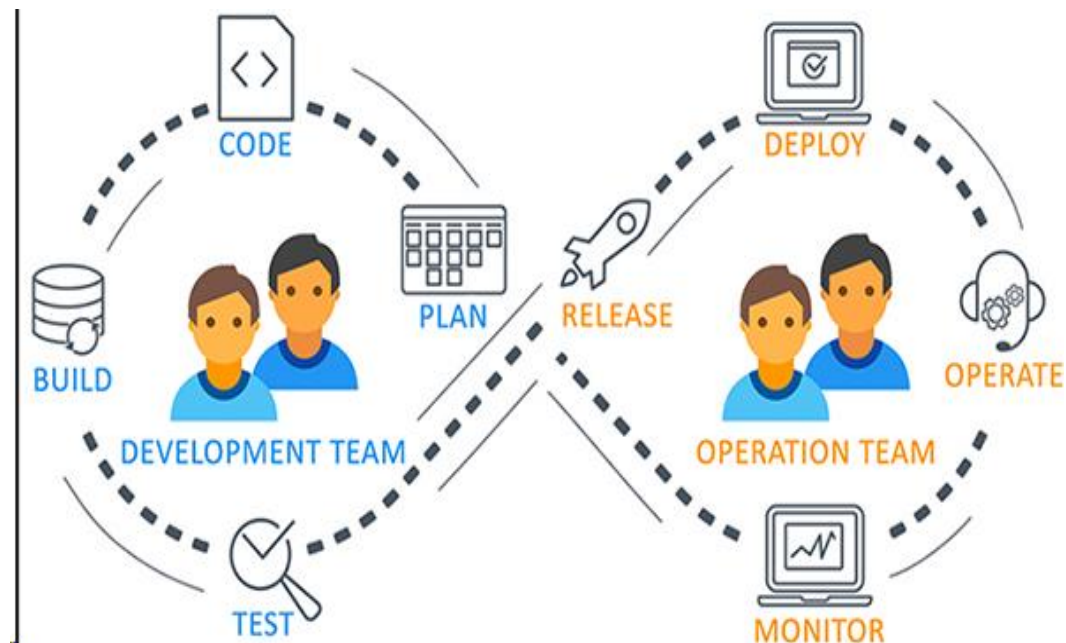


Devops:

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing**, **deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.



DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.

DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, **quality**, and **speed** of the application delivery has improved to a great extent.

DevOps is nothing but a practice or methodology of making "**Developers**" and "**Operations**" folks work together. DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach.

DevOps History

- In 2009, the first conference named **DevOps days** was held in Ghent Belgium. Belgian consultant and Patrick Debois founded the conference.
- In 2012, the state of DevOps report was launched and conceived by Alanna Brown at Puppet.
- In 2014, the annual State of DevOps report was published by Nicole Forsgren, Jez Humble, Gene Kim, and others. They found DevOps adoption was accelerating in 2014 also.

- In 2015, Nicole Forsgren, Gene Kim, and Jez Humble founded DORA (DevOps Research and Assignment).
- In 2017, Nicole Forsgren, Gene Kim, and Jez Humble published "Accelerate: Building and Scaling High Performing Technology Organizations".

Why Devops Is Important:

DevOps has become the gold standard in modern IT, but it means different things to different people. [What exactly is DevOps](#), and why has the demand for it increased so fast? DevOps is a collaboration between development and operation teams, which enables continuous delivery of applications and services to our end users. What makes DevOps stand out? Why is it chosen over the traditional method, and for what reasons? Let's go over DevOps fundamentals, understand why DevOps, and some of the tools why [DevOps engineers](#) are using today.

Why DevOps?: Before we get deep into what DevOps is and all the revolutions it brought with us, first understand why DevOps in the first place. and Before DevOps, there were two development models: Waterfall and Agile Method.

1. Waterfall Model

The waterfall model is the first model to be introduced in software development. It is a sequential process and very easy to understand. In this approach, software development is divided into several phases, and the output of one phase becomes the input for the next phase. This model is similar to a waterfall when the water flows off from the cliff; it cannot go back to its previous state.

The phases are; Requirements, Design, Implementation, Verification, and Maintenance.



Drawbacks of the waterfall model:

- It's difficult to make changes to the previous stage
- Not recommended for large-sized projects
- Developers and testers don't work together (which can result in a lot of bugs at the end)
- Not recommended for projects that will likely have changing requirements

From the figure below, we can see the issues with the waterfall model:

- The developer took a very long time to deploy code
- On the operations side, the tester found it challenging to identify problems and give useful feedback

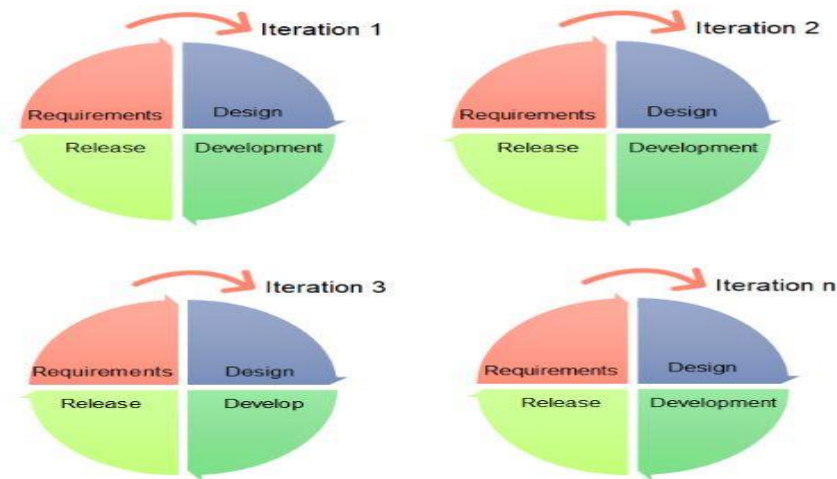


2. Agile Model

Agile is an approach in software development where each project splits into multiple iterations. As a result, at the end of each iteration, a software product is delivered. Each iteration lasts about one to three weeks. Every iteration involves functional teams working simultaneously on various areas, such as:

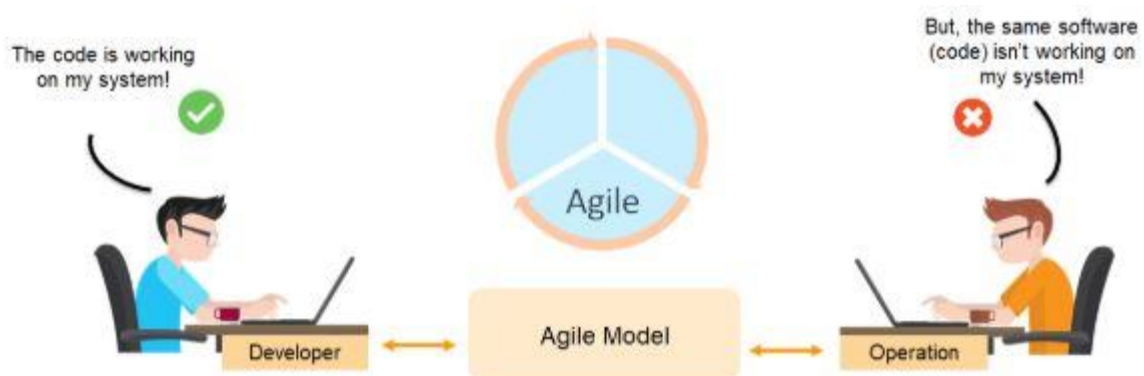
- Requirements
- Design
- Development
- Release

The figure below indicates that there can be a number of iterations needed to deliver a final product in agile method.



Using the agile method, the code that works for the developer may not work for the operations team.

So how can this issue be solved?



With DevOps, there is continuous integration between deployment of code and the testing of it. Near real-time monitoring and immediate feedback through a DevOps continuous monitoring tool enables both the developer and operations team work together.

The figure below shows how well the software is handled using DevOps.

