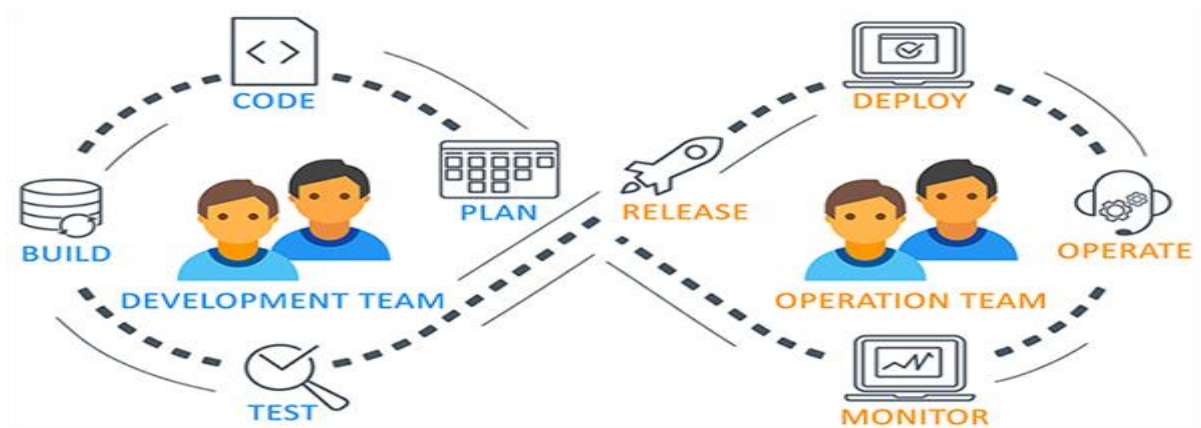


Lesson01: Introduction to DevOps

What is DevOps:

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from Agile methodology.

The concept of DevOps is related to software development, services and operations. Between the information operation personnel and that of software developers, DevOps helps in building communications, integration and collaboration.



DevOps History:

DevOps simply means the cross-department integration between Development, the department creating the code, and Operations, the department using that code.

The idea began in 2008 with a discussion between Patrick Debois and Andrew Clay Shafer concerning the concept of agile infrastructure. However, the idea only started to spread in 2009 with the advent of the first DevOpsDays event held in Belgium.

DevOps is lean thinking blended with agile philosophy. The revolutionary aspect of DevOps is that it crosses a traditional line by merging software development with the environment in which it is developed. This is more than just an attempt at efficiency, it is a cultural change. This cultural change is then made possible by a series of tools that automate such processes as code development and review through continuous integration, while allowing for version control.

Lean: The essence of Lean is to deliver value to the customer and to continuously improve the ability to deliver value by removing waste.

Lean Methodology:

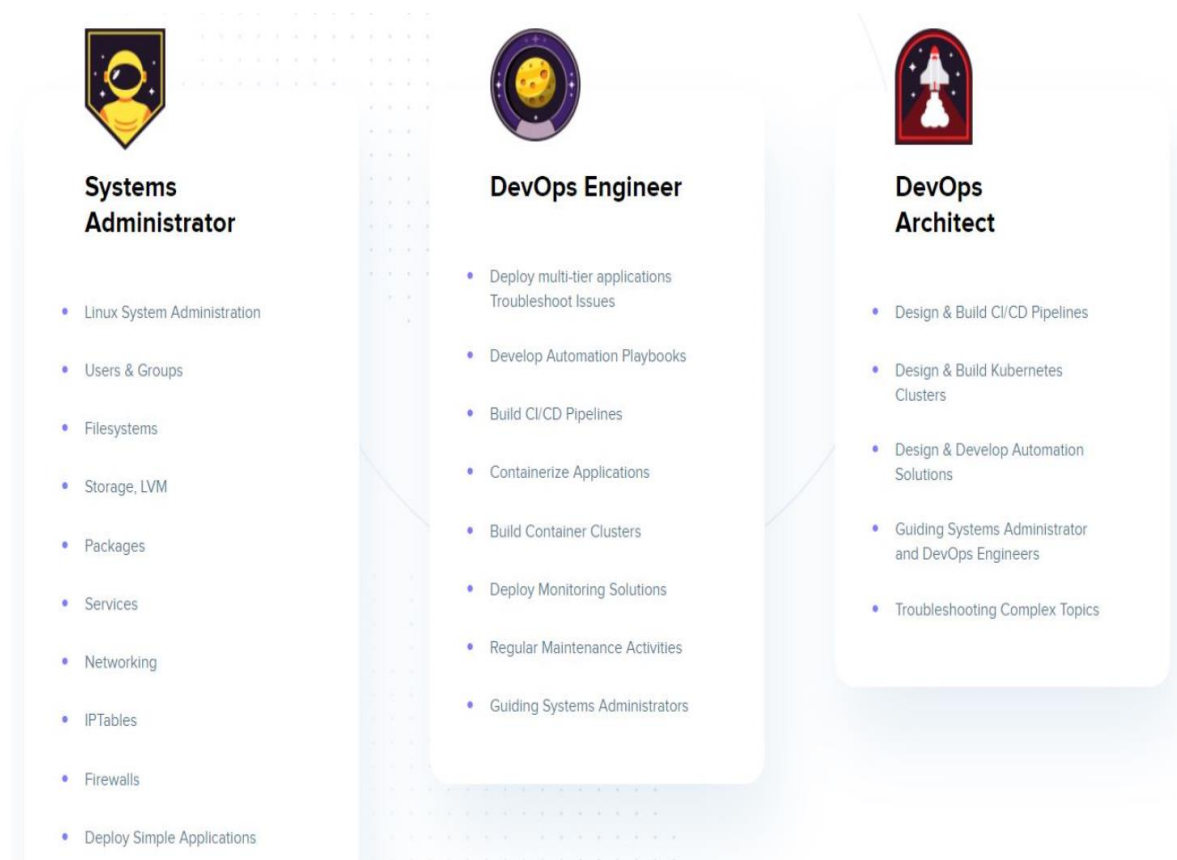
In short, Lean methodology is a way of optimizing the people, resources, effort, and energy of the organization toward creating value for the customer. It is based on two guiding tenets, continuous improvement and respect for people.

DevOps Engineer Roles and Responsibilities:

- Test, Build, Design, Deployment, and Ability to maintain continuous integration and continuous delivery process using DevOps tools.
- Must know how to choose the best tools and technologies which best fits the business needs.
- Ability to automate test and deploy the code and monitor.
- Work in close coordination with the development and operations team such that the application is in line with performance according to the customer's expectation.
- Good communicators: Since time immemorial, IT specialists have not been good communicators.
- Team player: It goes without saying that effective execution of DevOps strategy requires a combined effort of different individuals.
- Adaptive to change.
- Responsible on UX & Security feature.

Role & Responsibilities:

1. System Administrator.
2. DevOps Engineer.
3. DevOps Architect.



DevOps Culture to overcome legacy overhead:

One of the latest invention or trends in the world of software development is DevOps. In this field, the team of developers works with the operations staff to make sure that the software will run effectively and efficiently.

The topmost reasons why it is important to use DevOps are listed below:

- The technical benefit includes the delivery of software that too continuously. It has the benefit of having fewer complex problems to fix. It also has the benefit of faster and easier resolutions to every problem.
- The business-related benefits include faster delivery of features and it also has the advantage of having more stable operating environments. The benefits also include more time so that more value can be added.
- One of the important reasons is that DevOps provides the system for best communication. During the stage of production, there may arise certain problem related to blockage or delayed on progression. There are many teams involved during the production stage and to make a smooth collaboration between the teams DevOps is necessary. This will help you in having an idea about each prospect of the project.
- When there are bad developers, DevOps will slow them down. A company comprises of all kind of developers from excellent to very bad. The concept of DevOps will remove the bad developers since the bad codes that are produced by the developers will be left to be corrected.
- Also, the advantage of using DevOps is that it has significant shorter time to a market facility.
- The DevOps will also facilitate for the better quality of a product.
- Another advantage of using DevOps is that it has more reliable releases.
- Another great advantage is that it helps in improved productivity.
- The DevOps also helps in increasing the efficiency and improvement of customer satisfaction.

Release:

MAJOR: Major version when you make incompatible API changes.

MINOR: Minor version when you add functionality in a backwards compatible.

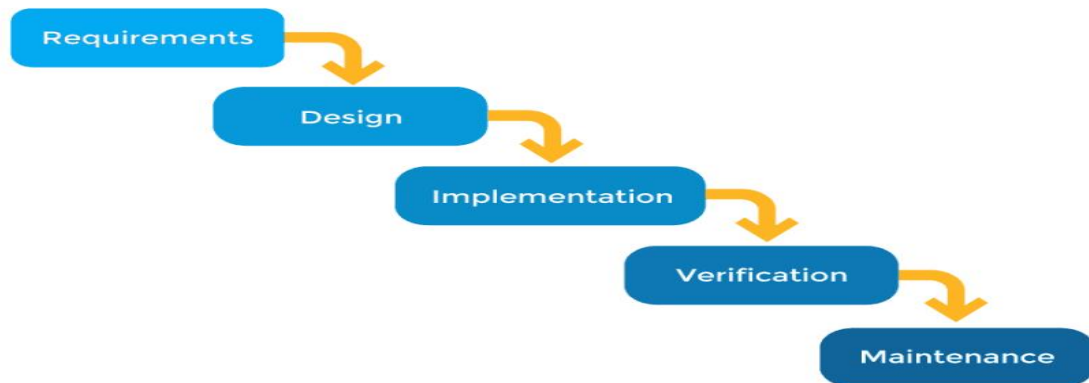
PATCH/Hot Fix: is version when you make backwards compatible bug fixes.

Meaning: 1.0.0.0

Software Development Life Cycle (SDLC):

- Waterfall Model:
- V-Model
- Prototyping Model
- Spiral Method
- Iterative and Incremental Method
- Agile Development
- DevOps

Waterfall Model:

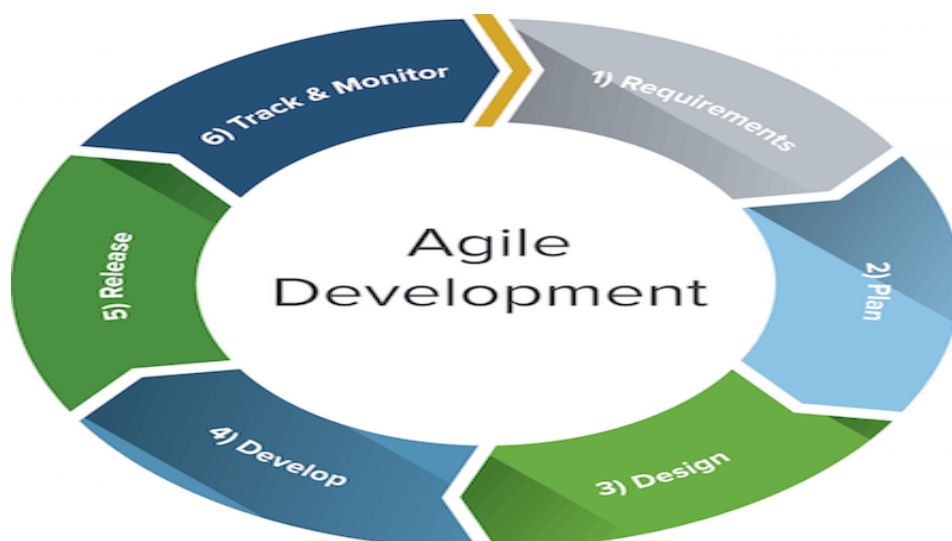


Though the waterfall model receives a lot of hate, certain problems are best solved with this methodology. For example, projects where the product has been clearly defined or projects that involve hardware upgrades can benefit from a waterfall approach.

there are disadvantages to this rigidity. First, we must assume that requirements can be frozen in place. This becomes more and more difficult with our ever-changing technical landscape.

Agile Development:

The agile method started with the creation of the "Manifesto for Agile Software Development" (now widely known as the Agile Manifesto) and its twelve principles.

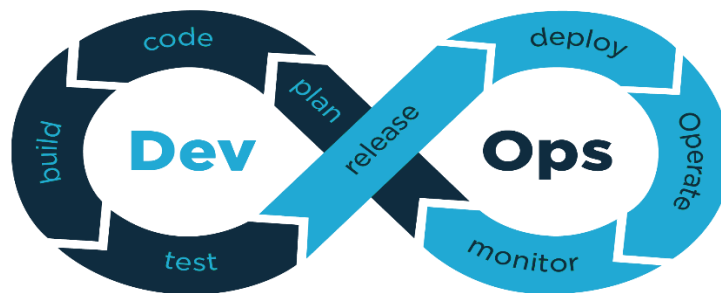


With this methodology, we focus on getting complete yet simple slices of functionality to our customers. We value communication, people, and early feedback and documentation usually takes a backseat to new features. With this methodology, changes in requirements don't cause problems. In fact, changes to requirements present opportunities. They're an expected part of the development of software.

However, with the way iterations fall, we typically can't estimate when a large project will finish. And scaling agile tends to be difficult, though scaled agile frameworks do provide relief.

DevOps:

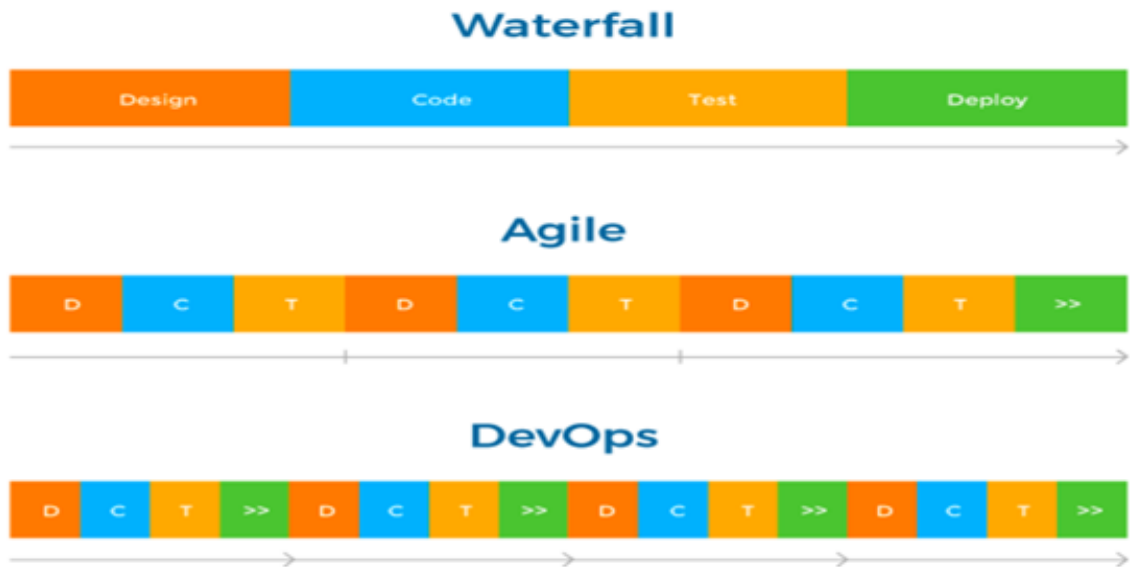
Let's welcome the newest methodology to the block: DevOps. DevOps brings development skills together with operations. This collaboration and sharing of responsibilities help ensure that the product developed operates well in production.



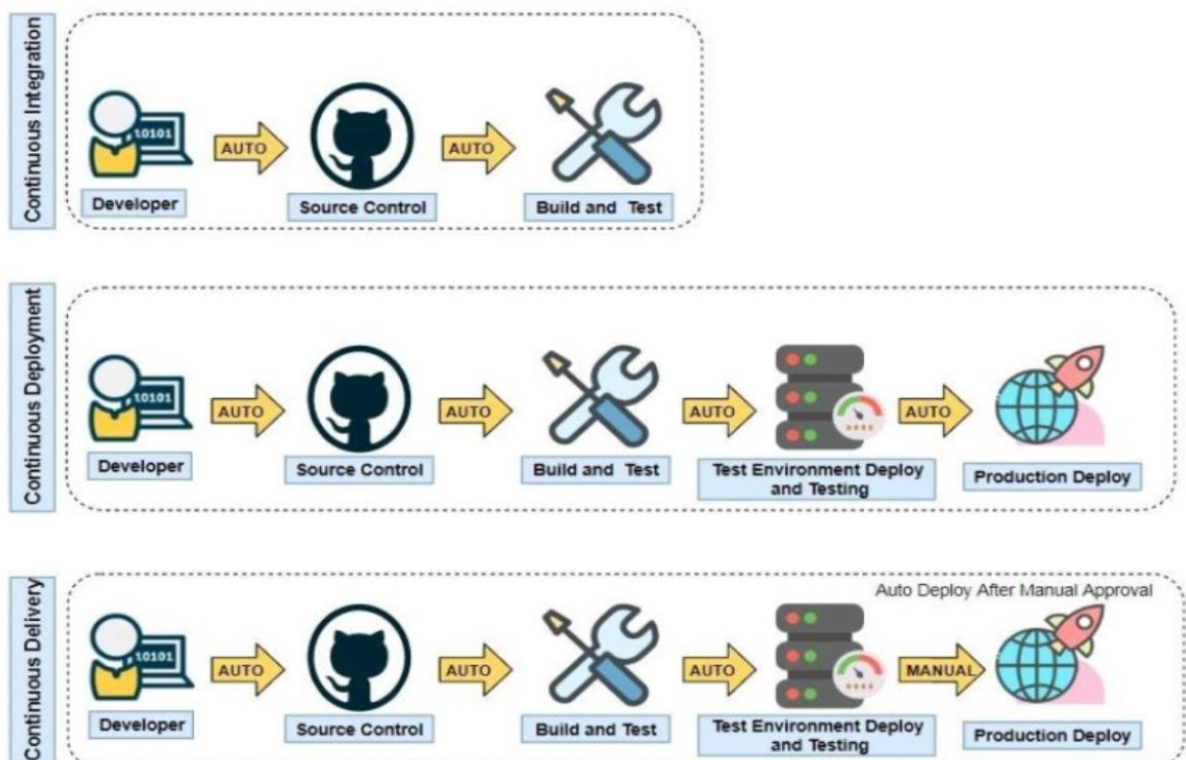
With this methodology, we not only work on the functional requirements, but we also automate operational requirements like monitoring and validating systems. We automate not only CI/CD but also the creation of production-like systems for development and testing. And we make tools so that software delivery teams can self-service their infrastructure needs.

The difficulty in making DevOps work revolves around experience. Most development teams have little experience with application or infrastructure monitoring. They don't always automate as much as possible. They might not feel confident in being responsible for the day-to-day operations of the software.

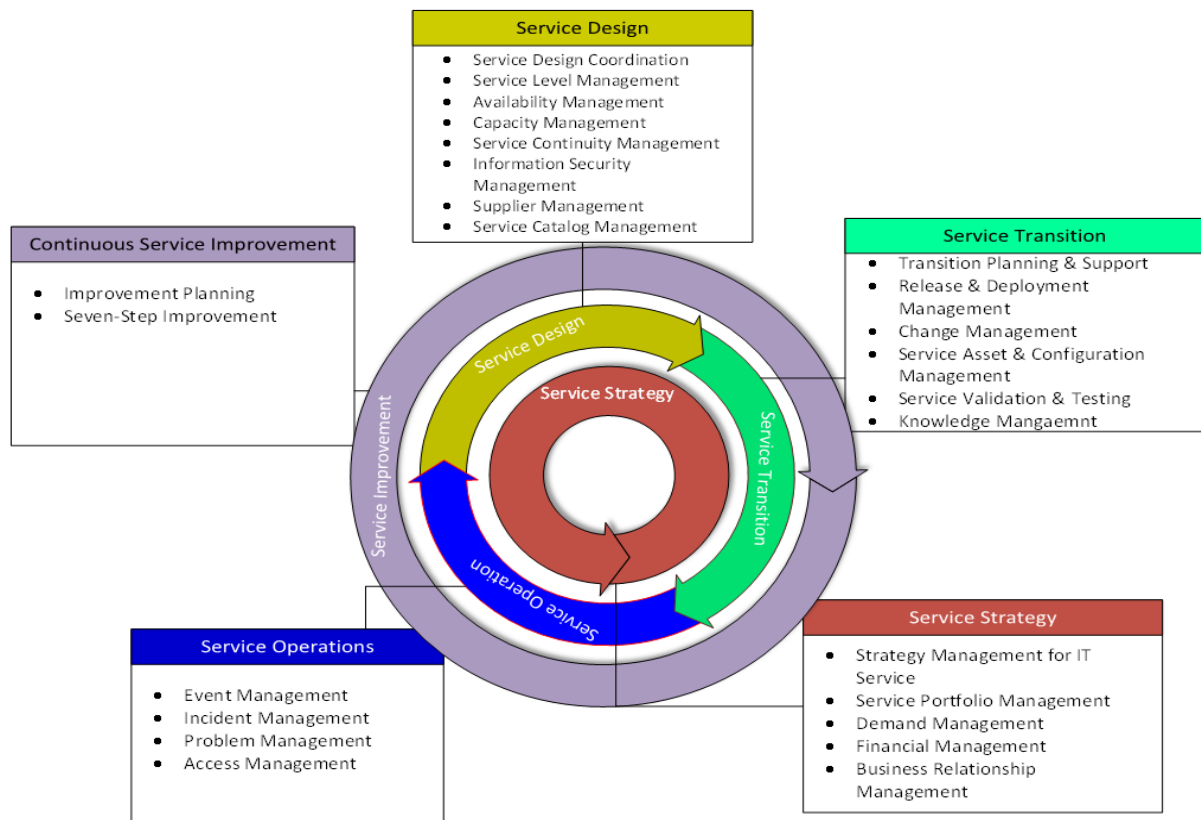
3 SDLC in a FRAME:



CI-CD Concept:



IT Service development and deployment process on ITIL:



Course Objective to Cover:

