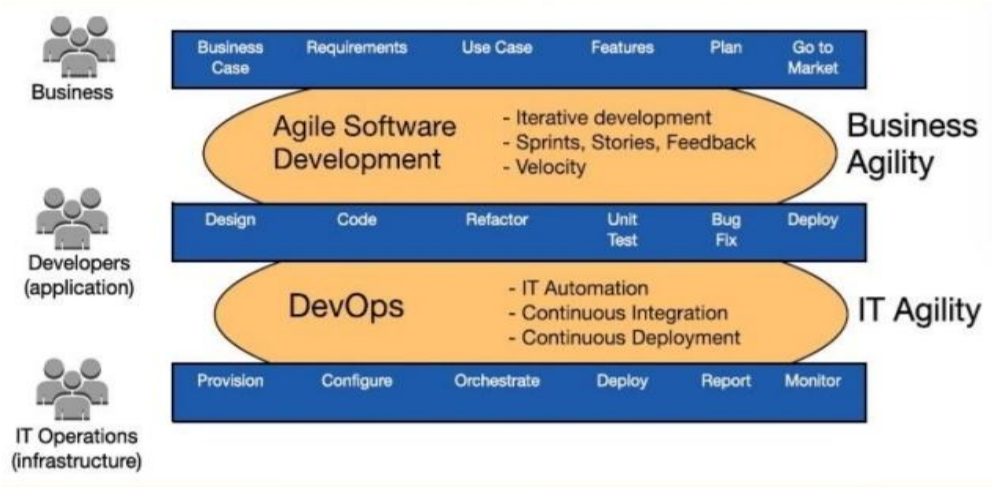
Introduction to DevOps

DevOps is a culture or an approach that encourages cross-functional collaboration between development and operations of software. It combines the principles of software development and software operations to develop software of better quality in less time with more reliability.

What is agile and is DevOps different from Agile?

Agile is a methodology that promotes continuous iteration of development and testing throughout the software development lifecycle of the project.

DevOps is a practice of bringing development and operations teams together whereas Agile is an iterative approach that focuses on collaboration, customer feedback and small rapid releases.



Agile addresses gap of communication between customers and developers while DevOps addresses gap of communication between developers and operations.

Core Values of DevOps

Culture

There's a shared understanding between developers and operations, and sharing of responsibility for the software they build. Since the teams are working together, it creates a positive attitude or culture among the teams which in turn improves the overall environment.

Automation

Automation helps to drive DevOps teams towards faster iteration and more stable builds. It frees up the team to focus more on important tasks that requires human knowledge. It helps to accelerate and enhance the benefits of DevOps as a whole.

Measurement
In DevOps, progress must be tracked. It can be tracked in form of customer satisfaction and feedback, deployment frequency, release frequency, etc.
Sharing
The Operations and Developers should be able to communicate with each other with transparency and openness. Tools, findings, defects, and experiences should be shared among the team.

DevOps Lifecycle

There are a total of seven phases in the DevOps lifecycle
1). Continuous Development
Plan- objectives
Code- requirements
First plan your application objectives that must be delivered to the customer, then start with project development. It includes development that involves code generation and putting the same to the next phase.

2). Continuous Integration
Plan- tests
Build- product
This includes several steps like the planning of tests that will be carried out in the next phase, understanding the code to produce the desired outcome as needed in the initial project documentation.

3). Continuous Testing
Test the product for actual usage in a live environment. The testing process gives more information about different aspects of an application that in turn is sent to the development process to improve the application.

4). Continuous Monitoring
Monitor the product usage and find out trends and the problem areas.

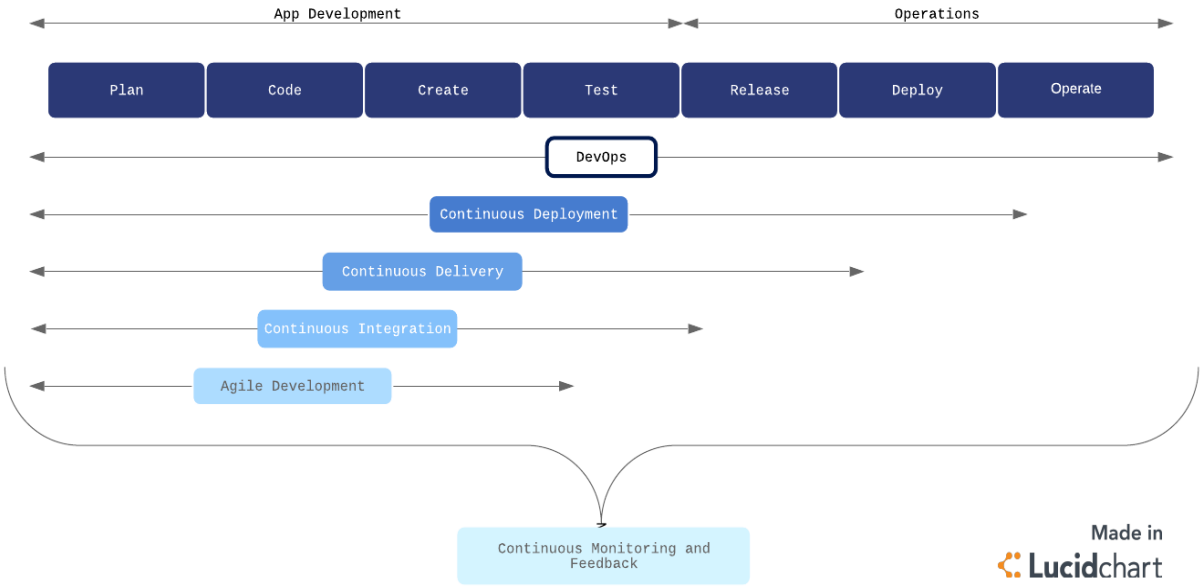
5). Continuous Feedback
Improve the current product with help of continuous customer feedbacks and release new versions quickly based on the response.

6). Continuous Deployment

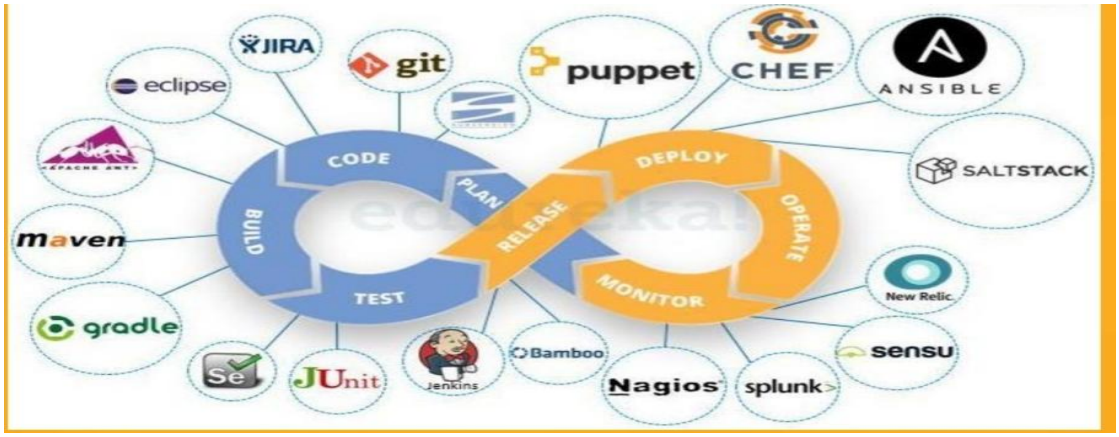
The deployment process is performed in such a way that any changes made in the code should not affect the functioning of high traffic website.

7). Continuous operations

All DevOps operations are based on continuity with complete automation of the release process
with shorter development cycles.



DevOps Tools



1. Collaboration Tools:
This type of tool is crucial to helping teams work together more easily, regardless of time zones or locations.
Eg. Slack

2. Planning Tools:
This type of tool is designed to provide transparency. Working together, teams can plan towards common goals, and better understanding of dependencies. Eg. Clarizen and Asana

3. Source Control Tools:
Source control tools provide an organized way to track changes over time. They manage the changes to the project, be it code, documents, environment configuration, documentation, database, compiled resources.
Eg. Git, Subversion

4. Issue Tracking Tools:
To track both internal as well as customer generated issues.These tools increase responsiveness and visibility.
Eg. Jira, ZenDesk

5. Configuration Management Tools:
Configuration management tools perform various roles to ensure consistency among physical and logical assets.
Eg.Puppet, Chef, Salt

6. Continuous Integration Tools:
These tools provide an immediate feedback loop by regularly merging code.
Eg.Jenkins, Bamboo

7. Automated Testing Tools:
These tools verify code quality before passing the build. Eg. Telerik, QTP, TestComplete

8. Deployment Tools:
Deployment tools streamline the process of distributing software and updates, usually via scheduling or automation. They also allow developers to collaborate on projects, track progress, and manage changes.
Eg. IBM uDeploy, CA Release Automation, XebiaLabs

Expected outcomes of DevOps

Velocity and speed of delivery to increase
Quality and stability will improve
Efficiency and capacity will increase.

Conclusion

DevOps has evolved as an extension of agile methodology and enables effective collaboration between developers and IT operation teams. DevOps methodology has many benefits such as ensures team collaboration, speed of delivery, reliability, security ensured with effective test automation practices.