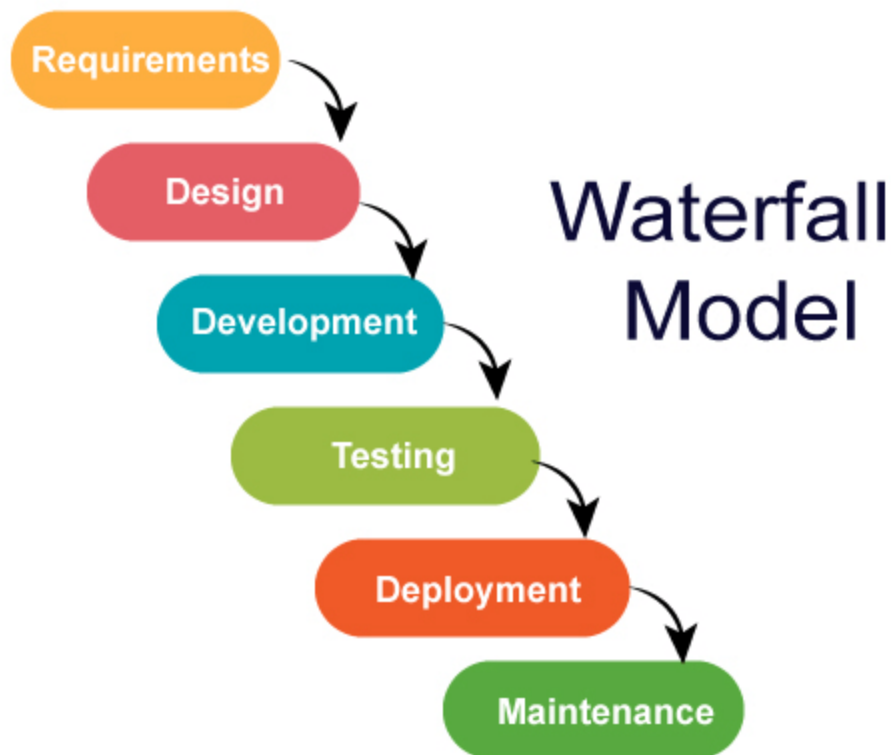


What is Devops?

Waterfall



Waterfall model is a traditional approach of software development. It works in a step by step manner in which we meet key requirements before moving to the next stage.



- Requirements: A client provides the specifications
- Design: Based on the specifications the best design is outlined and planned
- Development: The solution is built out by programmers
- Testing: Testing ensures the application is error-free and meets requirements
- Maintenance: The application is checked that it keeps working

Disadvantages

New requirements are hard to integrate



Any new requirements from the client are extremely difficult to re-integrate into the model. This means the solution may have to be restarted.

Poor feedback loops



The client will not see a solution for a long time, as it is being produced. Clients are unable to get their feedback in quickly and effectively.

Agile

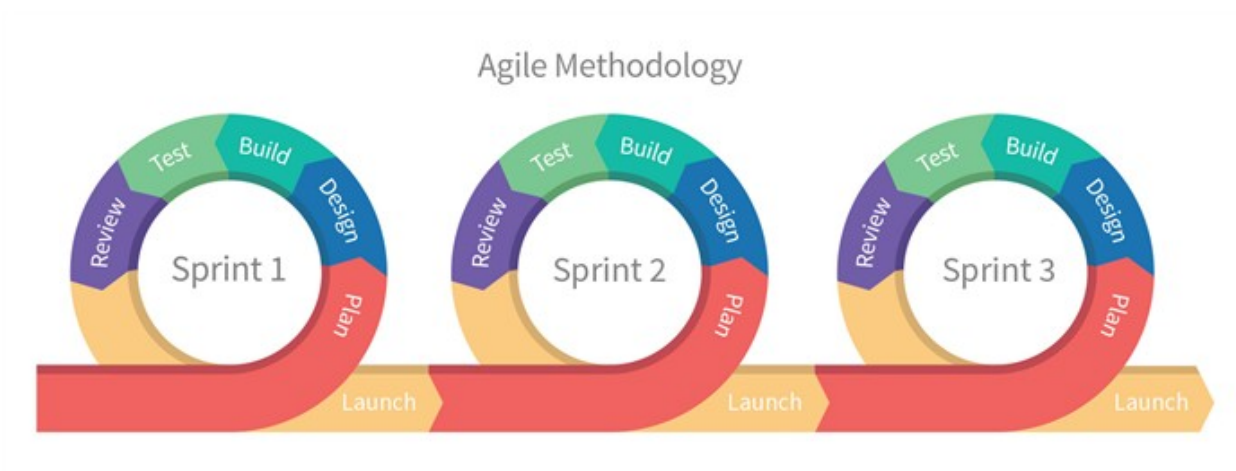


The weaknesses of the waterfall model led to the development of agile as a practice. It prioritizes quick feedback from clients and revolves around the development of prototypes. A programmer will create a solution, get immediate feedback and integrate the feedback. The entire process revolves around feedback loops.

Sprint approach



A sprint is an entire process of building a software into small actionable blocks. In which feedback is provided at the end of each sprint, and integrated into the next one.



- Plan
- Design
- Build
- Test
- Review

Backlog



A backlog are features that are put in a backlog for development in the future. These tasks are assigned based on priority and assigned to a sprint.

Advantages



Client requirements are better understood due to constant feedback. The product is also developed much quicker, in line with requirements and feedback.

Disadvantages



The product gets tested only on developer computers and not on production systems. Developers and operations team work in silos. This leads to developers sending their product to the operations team, which may be clueless to the flaws of the software due to siloed communication. These challenges that the operations team are faced with are due to lacking the knowledge of how the code works.

The operations team



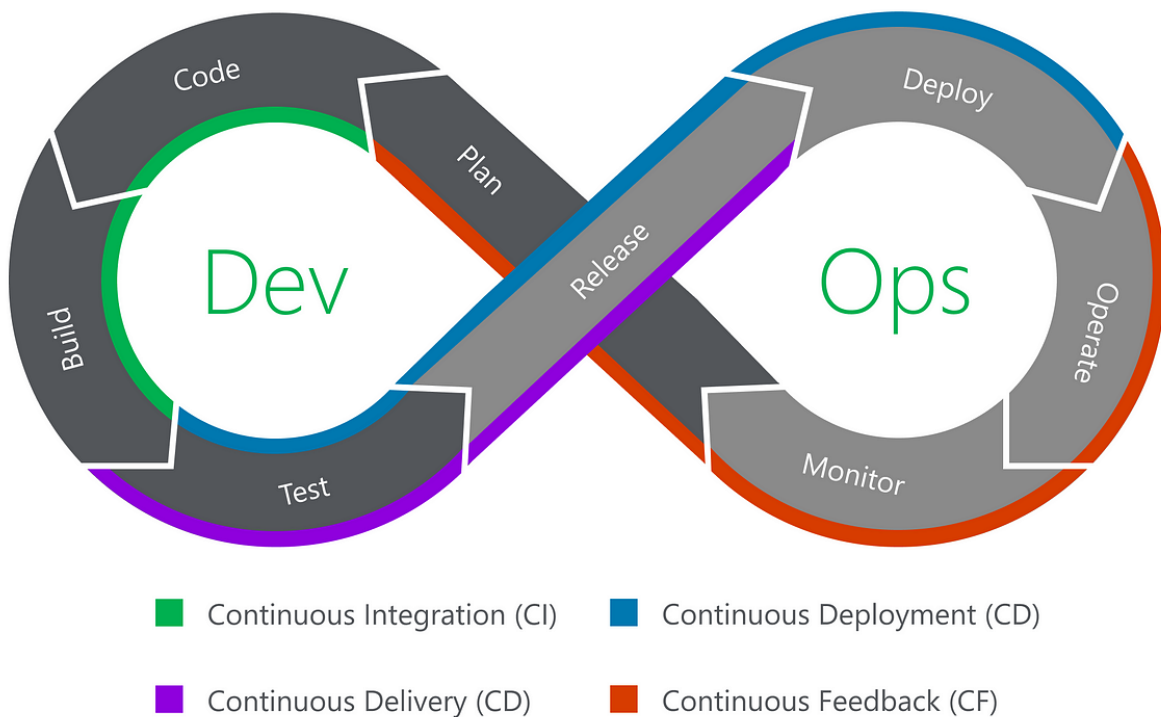
This team is responsible for ensure that the environment in which the product is running production is working smoothly.

DevOps (Developers and Operators)



DevOps is an evolution of the agile model of software development. The goal is to bring together the operations team and development team into a single team. **The idea is for developers to directly make software that are tested in a production environment. So that when it is deployed it will not have bugs.** The operations team can focus on analysing the production environment and providing feedback on what works well and poorly with the code.

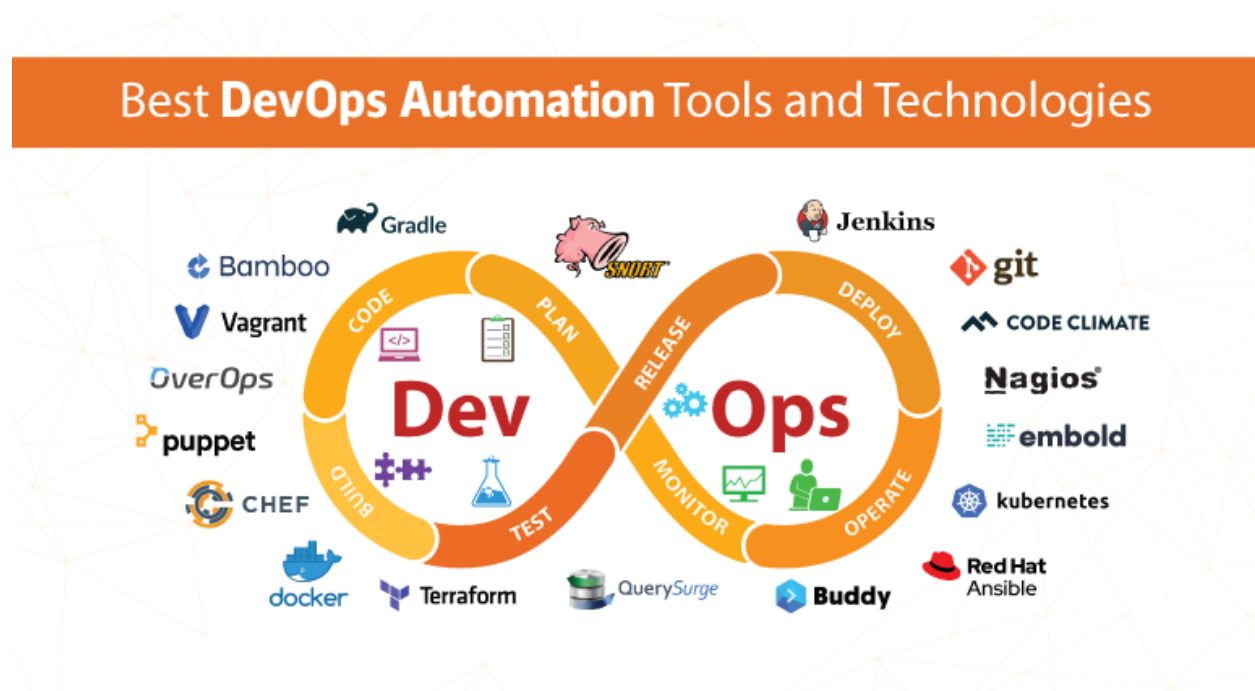
Communication, Collaboration and Security



- Planning: This is where the business owners and software development team discuss project goals and create a plan
- Code: Programmers then design and code the application and use tools like Git to store application code.
- Build: Use tools like maven and gravel to consistently build out your environment
- Test: The application is tested using automation testing tools like Selenium and JUnit to ensure software quality. Testing is scripted.
- Release/Integrate: When testing is complete, new features are integrated automatically to the already existing codebase. Tools like Jenkins allow for this.
- Deploy: Application is packaged after release and deployed from development server to production server

- Operate: Once software is deployed, operations team perform activities such as configuring servers and provisioning them with the required resources. Tools like Ansible, and Terraform come in.
- Monitoring: Monitoring allows IT organization to identify specific issues of specific releases and understand the impact on end-users.

DevOps Tools



A lot of these tools are open source and allow you to script your environment. The more you are able to script out your environment the more effective you can be at automating your processes.

Devops processes

Continuous Delivery



The continuous delivery process is the plan, code, build and test environment.

Continuous Integration

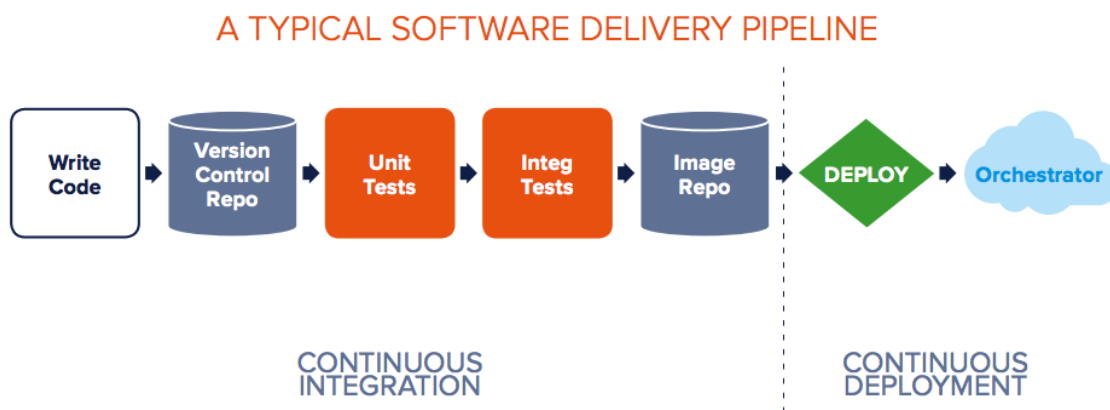


Once the continuous delivery stage is complete you want to be able to send your code to the build environment and test the code. Testing allows us to very quickly identify if there are any defects in the code.

Continuous Deployment



The next step if the tests are complete to push the code to production where it will be deployed, and monitored as it operates.



DevOps advantages

- Time taken to create and deliver software is reduced
- Complexity of maintaining an application is reduced
- Improved collaboration between developers and operations team
- Continuous integration and delivery ensure faster time to market