**EXP 1: In depth knowledge of devops methodology.**

Before DevOps, We had two approaches for software development namely the Waterfall and the Agile.

**Waterfall model:** The waterfall model is a software development model that is pretty straight forward and linear. This model follows a top-down approach.

Disadvantages of waterfall model:

* Risky and uncertain
* Lack of visibility of the current progress
* Not suitable when the requirements keep changing
* Difficult to make changes to the product when it is in the testing phase
* The end product is available only at the end of the cycle
* Not suitable for large and complex projects

**Agile Methodology**

Agile Methodology is an iterative based software development approach where the software project is broken down into various iterations or sprints. Each iteration has phases like the waterfall model such as Requirements Gathering, Design, Development, Testing, and Maintenance. The duration of each iteration is generally 2-8 weeks.

* Highly dependent on clear customer requirements
* Quite Difficult to predict time and effort for larger projects
* Not suitable for complex projects
* Lacks documentation efficiency
* Increased maintainability risks
* **SDLC: Software Development Life Cycle.**
* Development Team Operation Team

TEST

BUILD

CODE

PLAN

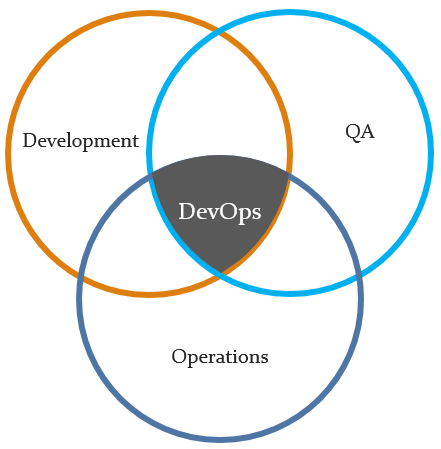
MAINTAIN

OPERATIONS

Deploy

**What is DevOps?**

**DevOps** is a software development methodology that combines software development (**Dev**) with information technology operations (**Ops**) participating together in the entire service lifecycle, from design through the development process to production support.



**The goals of DevOps**

* Fast Development Methodologies
* Fast Quality Assurance Methodologies
* Fast Deployment Methodologies
* Faster time to market
* Iteration & Continuous Feedback (strong and continuous communication between stakeholders — the end users and customers, product owners, development, quality assurance, and production engineers)

**The benefits of DevOps**

***Environment Stabilization****: Enforces consistency, increase up-time*

***Shorter Development Cycle****: Manage requirements and code-repository*

***Increased Release Velocity****: Continuous build, push-button deployments*

***Reduced Defects****: Regiment processes, automated testing*

***Process Metrics****: Track both time at each stage, and the errors and exceptions*

**The steps of DevOps**

* **Plan**: task management, schedules
* **Code**: code development and code review, source code management tools, code merging
* **Build**: continuous integration tools, version control tools, build status
* **Test**: continuous testing tools that provide feedback on business risks, determine performance
* **Package**: artifact repository, application pre-deployment staging
* **Release**: change management, release approvals, release automation
* **Operate**: infrastructure installation, infrastructure changes (scalability), infrastructure configuration and management, infrastructure as code tools, capacity planning, capacity & resource management, security check, service deployment, high availability (HA), data recovery, log/backup management, database management
* **Monitor**: service performance monitoring, log monitoring, end user experience, incident management

