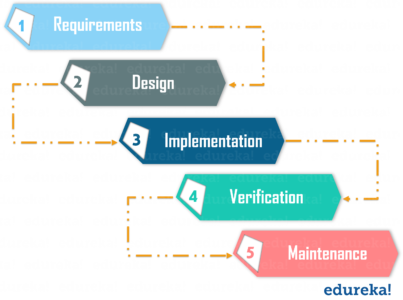
**Waterfall Model**

* The waterfall model is a software development model that is pretty straight forward and linear. This model follows a top-down approach.
* This model has various phases starting with **Requirements gathering and analysis**. This is the phase where you get the requirements from the client for developing an application. After this, you try to analyze these requirements.

****

* The next phase is the **Design** phase where you prepare a blueprint of the software. Here, you think about how the software is actually going to look like.
* Once the design is ready, you move further with the **Implementation** phase where you begin with the coding for the application. The team of developers works together on various components of the application.
* Once you complete the application development, you test it in the **Verification** phase. There are various tests conducted on the application such as unit testing, integration testing, performance testing, etc.
* After all the tests on the application are completed, it is deployed onto the production servers.
* At last, comes the **Maintenance** phase. In this phase, the application is monitored for performance. Any issues related to the performance of the application are resolved in this phase.

**Advantages of the Waterfall Model:**

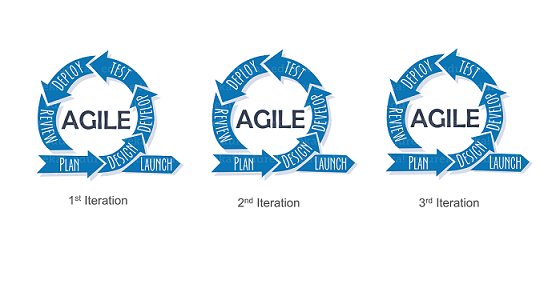
* Simple to understand and use
* Allows for easy testing and analysis
* Good for small projects if all requirements are clearly defined

**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](https://www.edureka.co/blog/what-is-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.



#### ****Agile Process****

* In Agile, a company releases an application with some high-priority features in the first iteration.
* After its release, the end-users or the customers give you feedback about the performance of the application.
* Then you make the necessary changes to the application along with some new features and the application is again released which is the second iteration.
* You repeat this entire procedure until you achieve the desired software quality.

#### ****Advantages of Agile Model****

* It adaptively responds to requirement changes favorably
* Fixing errors early in the development process makes this process more cost-effective
* Improves the quality of the product and makes it highly error-free
* Allows for direct communication between people involved in software project
* Highly suitable for large & long-term projects
* Minimum resource requirements & very easy to manage

#### ****Disadvantages of Agile Model****

* 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

**When You Should Use Waterfall and When to Use Agile**

Use **Waterfall** if :

* You know that there will be no change in the scope and your work involves fixed-price contracts
* The project is very simple or you’ve done it many times before
* You know very well that the requirements are fixed.
* Customers know exactly what they want in advance
* You’re working with orderly and predictable projects

And use **Agile** if:

* There is no clear definition of the final product.
* The clients/stakeholders are capable enough to modify the scope
* You anticipate any kind of changes during the project
* Rapid deployment is the goal

## **What is DevOps?**DevOps Definition - What is DevOps - Edureka

* DevOps is a practice that allows a single team to manage the entire application development life cycle: development, testing, deployment, and monitoring.
* The ultimate goal of DevOps is to decrease the duration of the system’s development life cycle while delivering features, fixes, and updates frequently in close synchronization with business objectives.
* It helps to develop superior quality software quickly and with more reliability.
* It consists of various stages such as continuous development, integration, testing, deployment, and monitoring.

**What Does DevOps Do?**

* Integrates developers and operations teams
* Improve collaboration and productivity by:
  + Automating infrastructure
  + Automating workflows
  + Continuous measuring application performance

**Benefits**: -

* Cost avoidance
* Accelerated innovation
* Flexibility
* Agility & Speed

**DevOps lifecycle: -**

* Plan: - First stage of DevOps cycle, where you Plan, Track, Visualize and Summarize your Project before working/starting it.
* Code:- Where developers write their code
* Build:- Build is a pre-release version and is identified by a build number, rather than by a release number.
* Test:- Process of executing automated test.
* Release:- This phase helps to integrate code into a shared repository.
* Deploy:- Manage and maintain development and deployment of software system and servers in any computational environment.
* Operate:- This phase is to keep the system upgraded with the latest update.
* Monitor:- It ensures that the application is performing as desired and the environment is stable.

**Continuous Integration (CI):-** DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

CI = Code + build + test

**Continuous Delivery (CD):-**

* Continuous Delivery (CD) is a DevOps practice that refers to the building, testing, and delivering improvements to the software code.
* The phase is referred to as the extension of the Continuous Integration phase to make sure that new changes can be released to the customers quickly in a substantial manner.

CD = Code + build + test + delivery/release/packaging

**Continuous Deployment (CD):-**

* Continuous deployment is a strategy/practice in software development where code changes to an application are released automatically into the production environment.

CD = Code + build + test + packaging/delivery/release + deploy on production server.