**What is Software Development**?

Software development is the process of creating computer programs or applications that perform specific tasks or solve particular problems. It involves designing, coding, testing, and maintaining software to meet the needs and requirements of users or businesses.

In simple terms, software is a set of instructions or programs that tell a computer (electronic device) what to do.

Software development is the process of transforming customer requirements into software applications.

**What is the Software Development Lifecycle (SDLC)** ?

The SDLC provides a structured way to guide software development projects, ensuring that they are well-planned, executed, and maintained.

**Types of SDLC**

1. Waterfall model
2. Agile model
3. DevOps

**Waterfall Model**:

Waterfall model is a sequential, linear process. It consists of several discrete phases. No phase begins until the prior phase is complete. The waterfall model does not allow you to return to a previous phase. The only way to revisit a phase is to start over at phase one.



**Feasibility check**: In the waterfall model, a feasibility study is conducted during the initial phase of the project to determine whether it is practical and viable to proceed with the software development. It assesses various factors to evaluate the project's feasibility and potential for success.

**Requirement Analysis**: The requirement analysis phase in the waterfall model is critical for setting the foundation of the software development project. It helps establish a clear understanding of the software's purpose, features, and constraints. A thorough and well-executed requirement analysis lays the groundwork for subsequent phases such as design, development, and testing, ensuring that the software aligns with stakeholders' expectations and delivers the desired outcomes.

**Design**:

The design phase in the waterfall model provides a detailed and comprehensive plan for the software development process. It guides the development team by defining the system's structure, internal workings, and user interface. The design phase acts as a reference point for developers to implement the software according to the specified design specifications, ensuring that the final product meets the requirements and objectives established in earlier phases.

**Coding and Unit Test**:

In this phase, the developers write the actual code for the software based on the design specifications. Code is reviewed and tested by the developers themselves to identify any errors or bugs. It ensures that the software functions as expected on a smaller scale before moving on to integration and system testing in later phases.

**Integration and System Testing:**

In this phase, the individual modules developed during the coding phase are integrated to form a complete software. This involves combining the code. Once the software is integrated, testing takes place on the whole combined software.

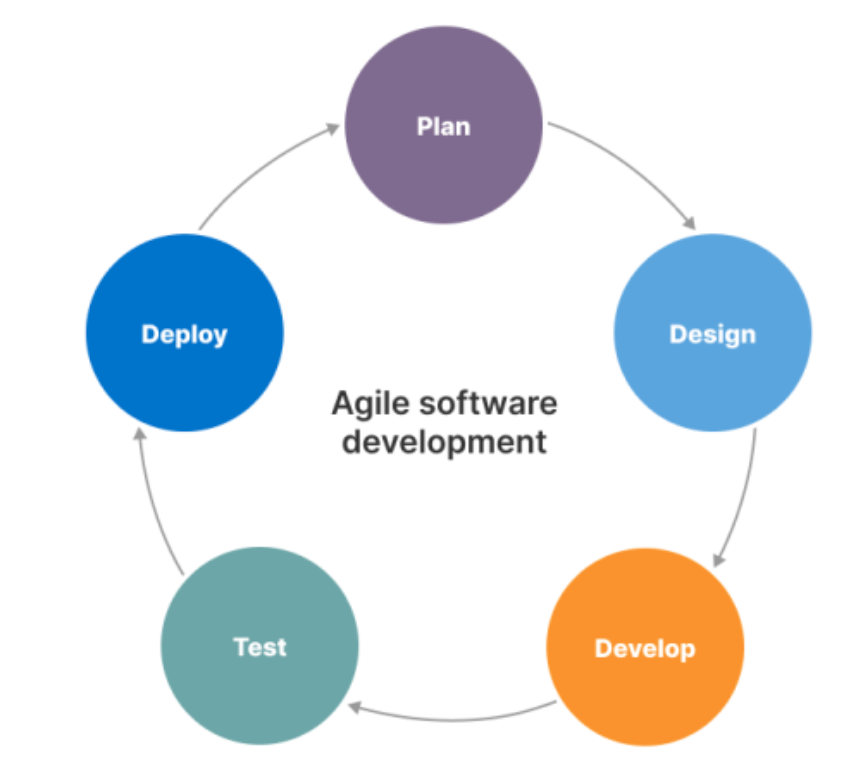
Maintenance: After software is given to the client and deployed in the client environment, this phase focuses on giving support to clients for any issues in software.

**Drawback of waterfall model**:

The drawbacks of the waterfall model include inflexibility, limited user involvement, risk of requirement misinterpretation, late problem detection and longer time to deliver value, no new feature addition. These limitations stem from its sequential nature, lack of flexibility, and limited opportunities for feedback and adaptation. As a result, more iterative and agile methodologies have gained popularity in modern software development.

**Agile Model**

The Agile model is an iterative and flexible approach to software development that emphasizes team collaboration, change adaptability, and continuous improvement. It involves breaking the development process into small chunks in incremental iterations called sprints, where multiple teams work collaboratively to deliver working software in short cycles.



Basically all features are not developed and deployed together, instead few features are taken first, as in diagram, planning, designing, development, testing and deployment is followed. Then it comes back for other features and repeats the cycle.

**Advantages of Agile**:

1. Enhanced customer satisfaction through frequent and early delivery of working software.
2. Increased adaptability to changing requirements.
3. Improved collaboration and communication within multiple teams.
4. Faster time to deploy and shorter cycles.
5. Continuous feedback and improvement, leading to higher-quality software

**Why devops?**

Software development and deployment were often manual and time-consuming, involving manual configuration, deployment, and testing. This manual approach increased the risk of errors, slowed down the release cycles, and hindered scalability.

The lack of collaboration and communication between development and operations teams resulted in blame games, finger-pointing, and a lack of shared goals.

Development and production environments often differed, causing deployment issues and making it challenging to replicate and debug problems. This led to increased deployment failures and instability in production.

**DevOps**

DevOps is a software development approach that combines development (Dev) and operations (Ops) teams to streamline collaboration, automate processes, and achieve continuous delivery. It promotes a culture of shared responsibility, efficient communication, automation, and rapid iteration to deliver high-quality software faster and more reliably.