There are numerous SDLC (Software Development Life Cycle) models that are followed during the software development phase. Each model follows a series of phases unique to its type to ensure success in the step of software development.

2 such models are agile model and waterfall model.

Agile Model

The Agile methodology promotes the continuous interaction of the development and testing during the SDLC process of any project. Unlike the waterfall model, the development and testing activities in the agile model are simultaneous. Agile methodology allows much communication between the customers, developers, testers, and managers.

In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.



Agile development methodology and testing practices have worked wonders for several organizations with positive aspects. The positive aspects of agile are not hidden. They are very much visible in organizations. There are some of the important points related to the agile model listed as follows –

* Agile focuses on customer feedback, collaboration, small and rapid releases.
* Its purpose is to manage complex projects.
* The Agile produces better application suites with the desired requirements. Moreover, it can quickly adapt according to the changes made on time during the project life.
* It has a small team size. Therefore, fewer people work on it so that they can move faster.
* The agile model is not a suitable model for small projects. The expenses of developing the small projects using agile are more than compared to other models.
* In agile methodology, the interaction of customers is very high, as after each iteration an incremental model is deployed to customers.

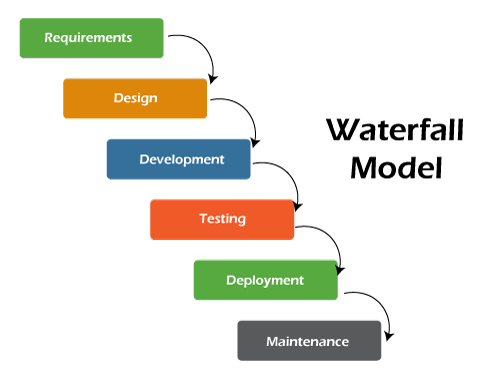
EXAMPLE:

Consider a company developing a mobile application, using the Scrum framework:

1. **Product Backlog Creation:** The product owner creates a prioritized list of features and user stories for the mobile app, known as the product backlog.
2. **Sprint Planning:** At the beginning of each sprint, the development team and product owner select a set of items from the product backlog to work on during the sprint. They break down these items into smaller tasks and estimate the effort required.
3. **Sprint Execution:** During the sprint, the development team works on implementing the selected features and user stories. They have daily stand-up meetings to discuss progress, challenges, and plan their work for the next 24 hours.
4. **Sprint Review:** At the end of the sprint, the team demonstrates the completed work to the product owner and stakeholders. They gather feedback and discuss any adjustments needed to the product backlog.
5. **Sprint Retrospective:** After the sprint review, the team holds a retrospective meeting to reflect on their process and identify ways to improve. They discuss what went well, what could be improved, and any action items for the next sprint.
6. **Repeat:** The process continues with subsequent sprints, each delivering additional functionality to the mobile app based on the evolving needs and priorities of the stakeholders.

## Waterfall model

It is one of the easiest and traditional model to manage. Because of its traditional development nature, each phase has specific deliverables and a review process. The waterfall model works well in smaller size projects where requirements are easily understandable.



The waterfall model is a universally accepted SDLC model. In this method, the whole process of software development is divided into various phases. The development in the waterfall model is seen as flowing steadily downwards (like a waterfall) as it is a continuous software development model. This model is named "Waterfall Model", because its diagrammatic representation resembles a cascade of waterfalls. Some important points related to the waterfall model are listed as follows -

* Waterfall model is not an ideal model to develop a large scale project size.
* The requirements in the waterfall model should be clear cut at the beginning time; otherwise, it may lead to a less effective method.
* In the waterfall model, it is hard to move back in order to make changes in the previous phase.
* The testing process in the waterfall model starts after the completion of development. So, there is a high chance of bugs to be found later in the project development.

EXAMPLE:

Imagine the same company is developing a new mobile application using the Waterfall model. Here's how the process might unfold:

1. **Requirements Phase:** The team gathers all the requirements for the mobile app, including features, functionalities, and user expectations. This phase might take several weeks of meetings and documentation.
2. **Design Phase:** Once the requirements are finalized, the design phase begins. The team creates detailed design documents specifying how the app will look and function. This includes user interface design, database design, and system architecture.
3. **Implementation Phase:** With the designs approved, development work starts based on the design specifications. Developers write code for the app based on the predefined requirements and design documents.
4. **Testing Phase:** After the development is complete, the testing team starts testing the application. They check for bugs, errors, and any deviations from the requirements. This phase can be time-consuming, especially if issues arise that require significant rework.
5. **Deployment Phase:** Once the testing is complete and the application is deemed ready, it is deployed to the production environment and made available to users.
6. **Maintenance Phase:** After deployment, the team provides ongoing support, addressing any issues that arise and releasing updates as necessary.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no.** | **Purpose** | **Agile model** | **Waterfall model** |
| **1.** | **Definition** | Agile model follows the incremental approach, where each incremental part is developed through iteration after every timebox. | Waterfall model follows a sequential design process. |
| **2.** | **Progress** | In the agile model, the measurement of progress is in terms of developed and delivered functionalities. | In the waterfall model, generally the measurement of success is in terms of completed and reviewed artifacts. |
| **3.** | **Nature** | Agile model is flexible as there is a possibility of changing the requirements even after starting the development process. | On the other hand, the waterfall model is rigid as it does not allow to modify the requirements once the development process starts. |
| **4.** | **Customer interaction** | In Agile model, there is a high customer interaction. It is because, after every iteration, an incremental version is deployed to the customer. | Customer interaction in waterfall model is very less. It is because, in a waterfall model, the product is delivered to the customer after overall development. |
| **5.** | **Team size** | It has a small team size. As smaller is the team, the fewer people work on it so that they can move faster. | In the waterfall model, the team may consist more members. |
| **6.** | **Suitability** | Agile model is not a suitable model for small projects. The expenses of developing the small projects using agile is more than compared to other models. | Waterfall model works well in smaller size projects where requirements are easily understandable. But waterfall model is not suitable for developing the large projects. |
| **7.** | **Test plan** | The test plan is reviewed after each sprint. | Test plan is reviewed after complete development. |
| **8.** | **Testing** | Testing team can take part in the requirements change phase without problems. | It is difficult for the testing team to initiate any change in needs. |