Waterfall Model

Meaning

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

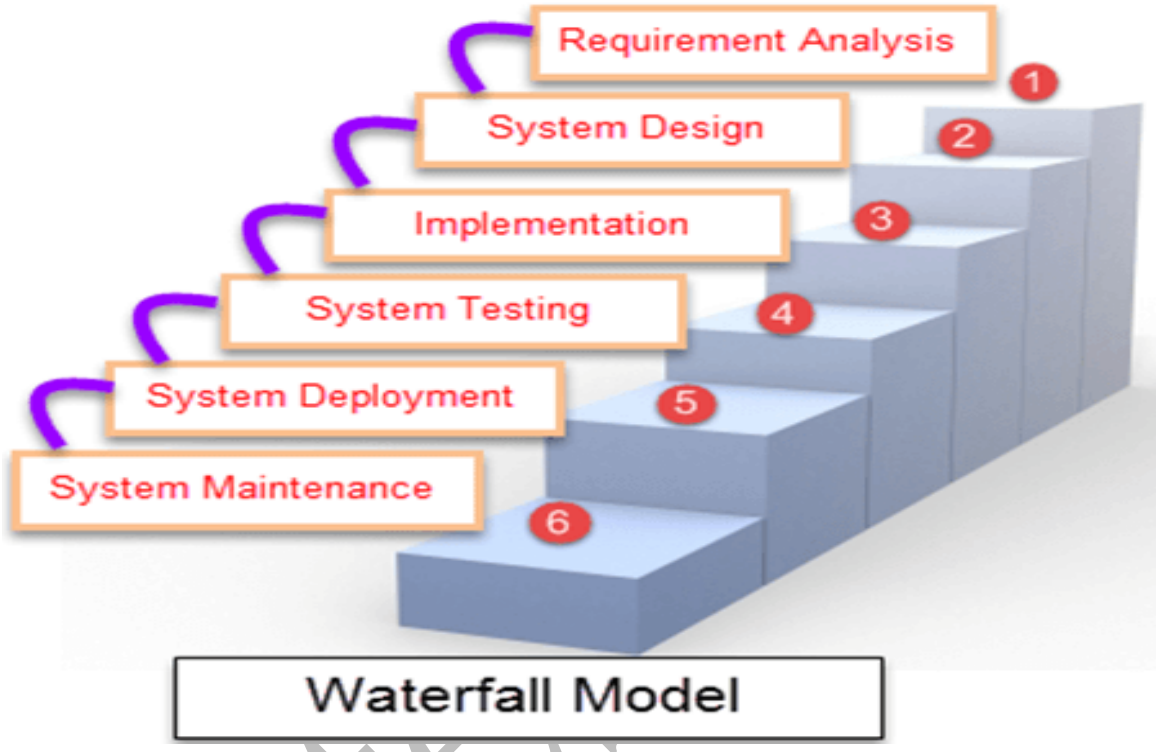
Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

- **Requirements:** The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.
- **System Design:** The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.

- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.



- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaws or errors. Testing is done so that the client does not face any problem during the installation of the software.

- **Deployment of System:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:** This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals is achieved for the previous phase and it is signed off, so the name "Waterfall Model".

Advantages of Waterfall Model

- Before the next phase of development, each phase must be completed
- Suited for smaller projects where requirements are well defined
- They should perform quality assurance test (Verification and Validation) before completing each stage
- Elaborate documentation is done at every phase of the software's development cycle
- Project is completely dependent on project team with minimum client intervention
- Any changes in software is made during the process of the development

Disadvantages of Waterfall Model

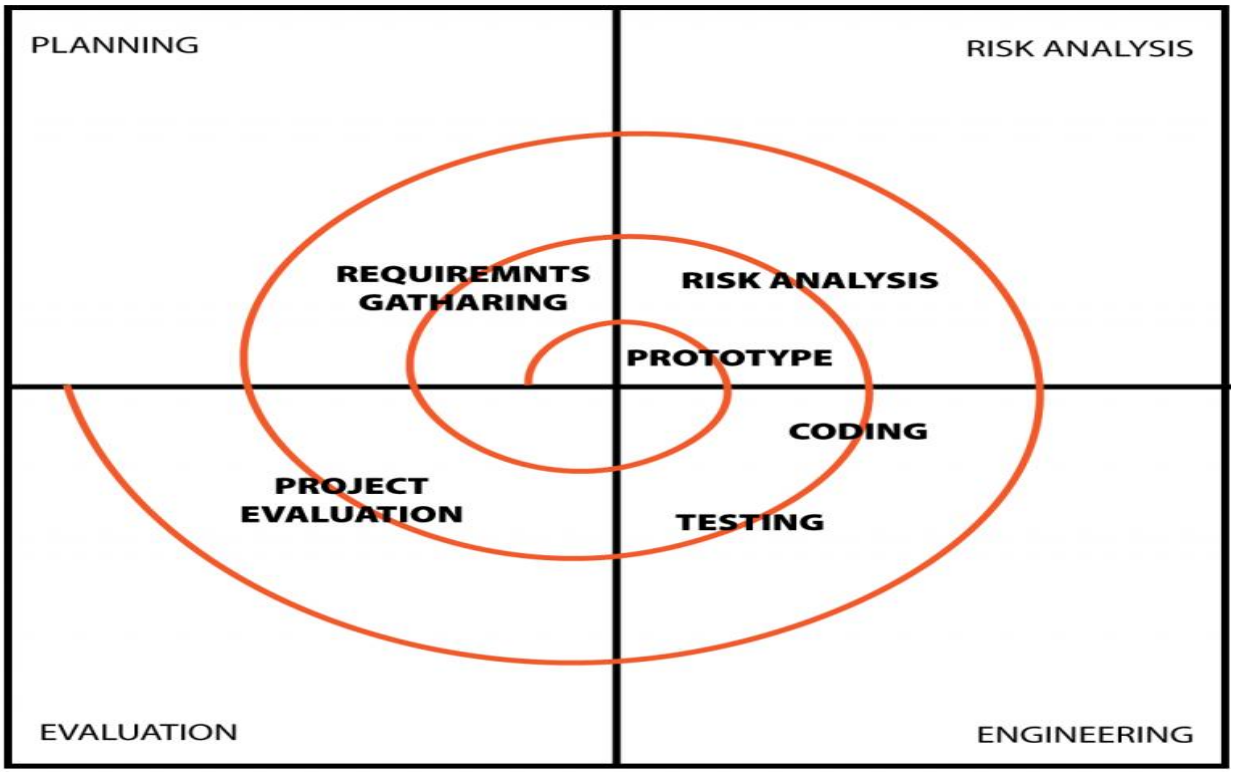
- Error can be fixed only during the phase
- It is not desirable for complex project where requirement changes frequently
- Testing period comes quite late in the developmental process
- Documentation occupies a lot of time of developers and testers
- Clients valuable feedback cannot be included with ongoing development phase
- Small changes or errors that arise in the completed software may cause a lot of problems

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Spiral Model



The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

Phases involved in Spiral Model

1. Planning phase: All the required information about the project will be gathered in this phase. Requirements such as BRS (business requirement specification and SRS (system requirement specifications), design alteration, etc. will be done in this phase. Cost estimation, scheduling the resources for iteration, etc. also happens in this phase.

2. Risk Analysis: Requirements of the project is studied and brainstorm sessions are conducted to figure out potential risks involved. Once the risk has been identified proper strategies and risk mitigation methodologies will be planned.

3. Testing phase: Testing alongside developmental changes will be done in this phase. Coding, test case development, test execution, test summary report, defect report generation, etc. happens in this phase.

4. Evaluation phase: Customer can evaluate the tests and can give feedback before the project goes to the next level

1st iteration – Activities such as panning, initial risk analysis, engineering evaluation, requirement gathering happens.

2nd iteration – Higher level planning, detailed risk analysis, evaluation happens in this phase

3rd iteration – Testing related activities such as coding, tool selection, resource allocation, which test to choose? Etc. happens in this phase.

4th iteration – In this customer is the key where they can evaluate the entire process and express their option regarding it.

When to use Spiral Methodology?

- When project is large
- When releases are required to be frequent
- When creation of a prototype is applicable
- When risk and costs evaluation is important
- For medium to high-risk projects
- When requirements are unclear and complex
- When changes may require at any time
- When long term project commitment is not feasible due to changes in economic priorities

Advantages of Spiral Model

The spiral model has some advantages compared to other SDLC models:

- **Suitable for large projects:** Spiral models are recommended when the project is large, bulky or complex to develop.
- **Risk Handling:** There are a lot of projects that have un-estimated risks involved with them. For such projects, the spiral model is the best SDLC model to pursue because it can analyze risk as well as handling risks at each phase of development.
- **Customer Satisfaction:** Customers can witness the development of product at every stage and thus, they can let themselves habituated with the system and throw feedbacks accordingly before the final product is made.
- **Requirements flexibility:** All the specific requirements needed at later stages can be included precisely if the development is done using this model.

Disadvantages of Spiral Model

- Cannot be used for small projects as it can be expensive
- A vast amount of documentation owing to several intermediate stages
- The end date of the project cannot be calculated at the early stages of the project
- Complex process
- High expertise is required to run the model