Google Custom Search SEARCH

Software Testing Help

Home Resources FREE EBooks QA Testing ▾ Courses ▾ Automation ▾
Types Of Testing ▾ Tutorials ▾

What Is SDLC Waterfall Model?

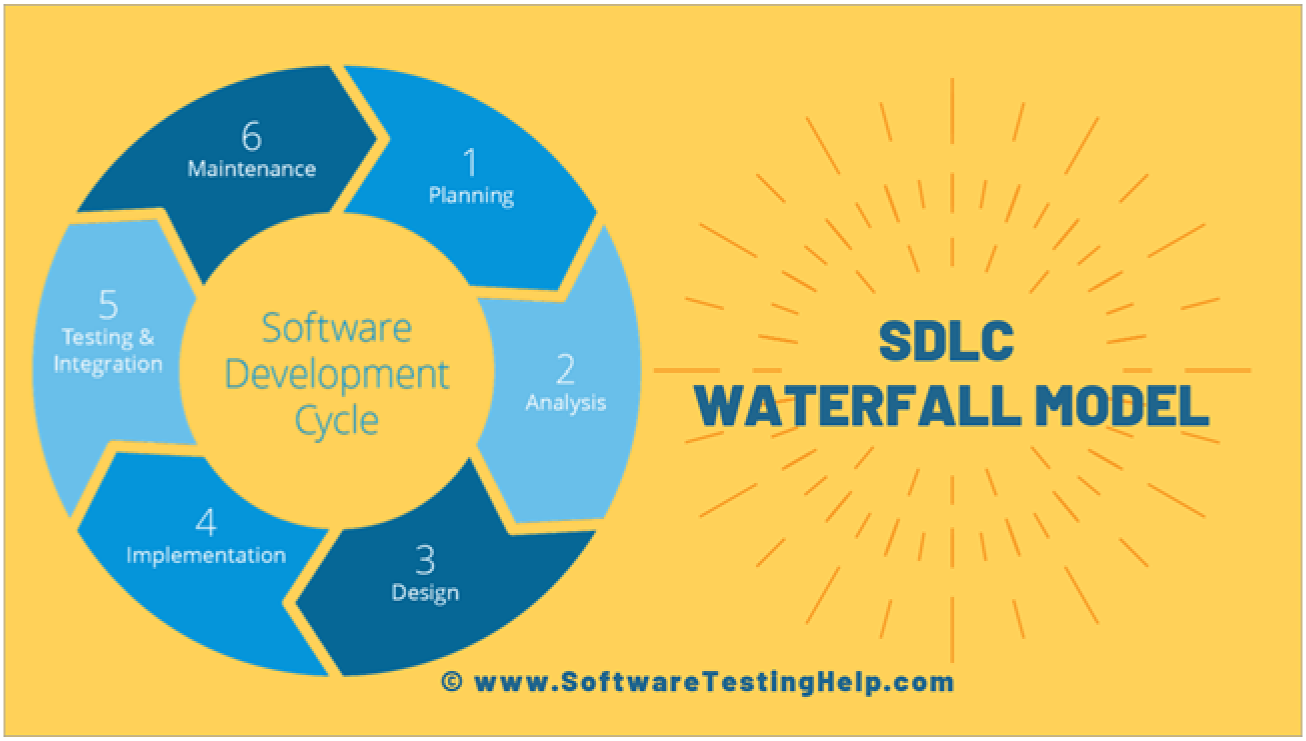
Last Updated: November 10, 2019

What is *SDLC Waterfall Model*?

Introduction:

Waterfall model is an example of a Sequential model. In this model, the software development activity is divided into different phases and each phase consists of a series of tasks and has different objectives.

Waterfall model is the pioneer of the SDLC processes. In fact, it was the first model which was widely used in the software industry. It is divided into phases and output of one phase becomes the input of the next phase. It is mandatory for a phase to be completed before the next phase starts. In short, there is no overlapping in the Waterfall model



In waterfall, development of one phase starts only when the previous phase is complete. Because of this nature, each phase of the waterfall model is quite precise well defined. Since the phases fall from a higher level to lower level, like a waterfall, It's named as the waterfall model.

Pictorial representation of the waterfall model:

The activities involved in different phases are as follows:

S.No	Phase	Activities Performed	Deliverables

S.No	Phase	Activities Performed	Deliverables
1	Requirement Analysis	1. Capture all the requirements. 2. Do brainstorming and walkthrough to understand the requirements. 3. Do the requirements feasibility test to ensure that the requirements are testable or not.	RUD (Requirements Understanding Document)
2	System Design	1. As per the requirements, create the design 2. Capture the hardware / software requirements. 3. Document the designs	HLD (High Level Design document) LLD (Low level design document)
3	Implementation	1. As per the design create the programes / code 2. Integrate the codes for the next phase. 3. Unit testing of the code	Programs Unit test cases and results
4	System Testing	1. Integrate the unit tested code and test it to make sure if it works as expected. 2. Perform all the testing activities (Functional and non functional) to make sure that the system meets the requirements. 3. In case of any anomaly, report it. 4. Track your progress on testing through tools like traceability metrics, ALM 5. Report your testing activities.	Test cases Test reports Defect reports Updated matrices.
5	System Deployment	1. Make sure that the environment is up 2. Make sure that there are no sev 1 defects open. 3. Make sure that the test exit criteria are met. 4. Deploy the application in the respective environment. 5. Perform a sanity check in the environment after the application is deployed to ensure the application does not break.	User Manual Environment definition / specification

S.No	Phase	Activities Performed	Deliverables
6	System maintenance	1. Make sure that the application is up and running in the respective environment. 2. Incase user encounters and defect, make sure to note and fix the issues faced. 3. Incase any issue is fixed; the updated code is deployed in the environment. 4.The application is always enhanced to incorporate more features, update the environment with the latest features	User Manual List of production tickets List of new features implemented.

What You Will Learn: [show]

When To Use SDLC Waterfall Model?

SDLC Waterfall model is used when

- Requirements are stable and not changed frequently.
- An application is small.
- There is no requirement which is not understood or not very clear.
- The environment is stable
- The tools and techniques used is stable and is not dynamic
- Resources are well trained and are available.

Pros and Cons of Waterfall model

Advantages of using the Waterfall model are as follows:

- Simple and easy to understand and use.
- For smaller projects, the waterfall model works well and yield the appropriate results.
- Since the phases are rigid and precise, one phase is done one at a time, it is easy to maintain.
- The entry and exit criteria are well defined, so it easy and systematic to proceed with quality.
- Results are well documented.

Disadvantages of using Waterfall model:

- Cannot adopt the changes in requirements
- It becomes very difficult to move back to the phase. For example, if the application has now moved to the testing stage and there is a change in requirement, It becomes difficult to go back and change it.

- Delivery of the final product is late as there is no prototype which is demonstrated intermediately.
- For bigger and complex projects, this model is not good as a risk factor is higher.
- Not suitable for the projects where requirements are changed frequently.
- Does not work for long and ongoing projects.
- Since the testing is done at a later stage, it does not allow identifying the challenges and risks in the earlier phase so the risk mitigation strategy is difficult to prepare.

Conclusion

In the waterfall model, it is very important to take the sign off of the deliverables of each phase. As of today most of the projects are moving with Agile and Prototype models, Waterfall model still holds good for smaller projects. If requirements are straightforward and testable, Waterfall model will yield the best results.

Recommended Reading

- Spiral Model - What is SDLC Spiral Model?
- SDLC (Software Development Life Cycle) Phases, Methodologies, Process, and Models
- Onsite - Offshore Model of Software Testing Projects (and How to Make It Work for You)
- Case Study: How to Eliminate Flaws of Waterfall and Agile Development Processes Using a Hybrid Model
- Zephyr Enterprise Test Management Tool Review - How to Use Waterfall Model Assets in Agile Tool
- Agile vs. Waterfall: Which is the Best Methodology for Your Project?
- Practical Software Testing - New FREE eBook [Download]
- What is STLC V-Model?

About SoftwareTestingHelp

Helping our community since 2006! Most popular portal for Software professionals with **100 million+ visits!** You will absolutely love our tutorials on Software Testing, Development, Software Reviews and much more!

Join Over 300,000+ Followers!

Top FREE Training Tutorials

- Load Testing Buyer’s Guide
- Appium Testing Tutorial
- Free QA Training
- Selenium Tutorials
- QTP/UFT Tutorials
- Quality Center QC Tutorials
- LoadRunner Tutorials
- JMeter Tutorials
- JIRA Tutorials
- VBScript Tutorials
- Best Test Management Tools
- Unix Tutorials
- DevOps Tutorials
- JAVA Tutorials
- Python Tutorials
- Free C++ Tutorials
- 101+ Interview Questions

Join Our Team!

ABOUT US | **CONTACT US** | **ADVERTISE** | **TESTING SERVICES**