

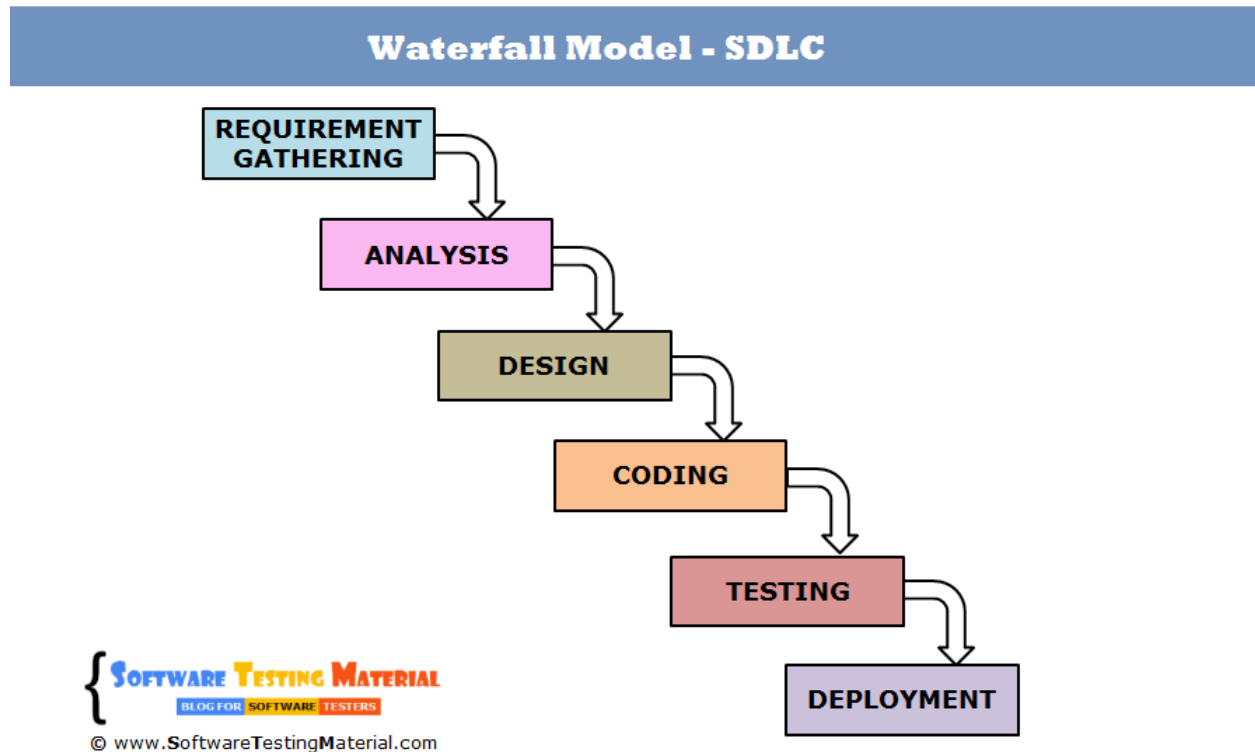
Waterfall Model in Software Development Life Cycle

April 5, 2016 by Rajkumar

Before start reading Waterfall Model, I would suggest you to check this post "[Software Development Life Cycle](#)"

You could see different types of Software Development Methodologies in that post. I have mentioned Waterfall Model as one of the Software Development Methodologies.

Let's see what is Waterfall Model and it's advantages and disadvantages in detail.



Waterfall Model:

Waterfall Model is a traditional model. It is aka Sequential Design Process, often used in [SDLC](#), in which the progress is seen as flowing downwards like a waterfall, through the different phases such as Requirement Gathering, Feasibility Study/Analysis, Design, Coding, Testing, Installation and Maintenance. Every next phase is begun only once the goal of previous phase is completed. This methodology is preferred in projects where quality is more important as compared to schedule or cost. This methodology is best suitable for short term projects where the requirements will not change. (E.g. Calculator, Attendance Management)

Advantages:

- Requirements do not change nor does design and code, so we get a stable product.
- This model is simple to implement. Requirements are finalized earlier in the life cycle. So there won't be any chaos in the next phases.
- Required resources to implement this model are minimal compared to other methodologies
- Every phase has specific [deliverable's](#). It gives high visibility to the project manager and clients about the progress of the project.

Disadvantages:

- Backtracking is not possible i.e., we cannot go back and change requirements once the design stage is reached.
- Change in requirements leads to change in design and code which results defects in the project due to overlapping of phases.
- Customer may not be satisfied, if the changes they required are not incorporated in the product.
- The end result of waterfall model may not be a flexible product
- In terms of testing, testers involve only in the testing phase. Requirements are not tested in the requirement phase. It can't be modified even though we identify that there is a bug in the requirement in the testing phase. It goes on till the end and leads to lot of re-work.

- It is not suitable for long term projects where requirements may change time to time
- Waterfall model can be used only when the requirements are very well known and fixed
-

Final words: Testing is not just finding bugs. As per the Waterfall Model, Testers involve only almost at the end of the [SDLC](#). Ages ago the mantra of testing is just to finding bugs in the software. Things changed a lot now. There are some other SDLC models implemented. I would post other models in the upcoming posts in detail with their advantages and disadvantages. It is up to your team to choose the SDLC model depends on the project you are working.