**ST. XAVIER’S COLLEGE**

**Maitighar, Kathmandu**



**Software Engineering**

**LAB ASSIGNMENT #4**

**Submitted by:**

**Aishwarya Rai**

**014BIM02**

**Submitted to:**

**Mr. Balkrishna Subedi**

**Department of Computer Science**

**St.Xavier’s College**

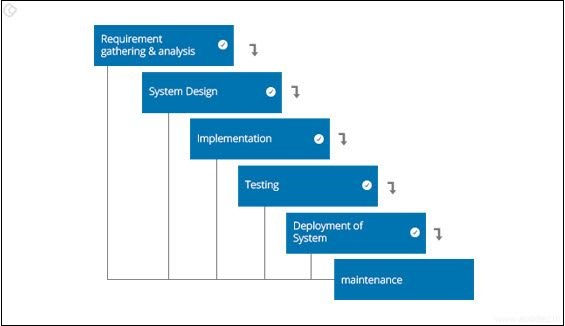
**WATERFALL MODEL**

The Waterfall Model was first Process Model to be introduced, also referred to as a linear-sequential life cycle model (unknown).The waterfall model describes a development method that is linear and sequential.

In a waterfall model, each phase must be completed fully before the next phase can begin. This type of [software development model](http://istqbexamcertification.com/what-are-the-software-development-models/) is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model [software testing](http://istqbexamcertification.com/what-is-a-software-testing/) starts only after the development is complete. In **waterfall model phases** do not OVERLAP.

Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

**Diagram:**



(Acodez)

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

The sequential phases in Waterfall model are −

**Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

**System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

**Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in 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. Post integration the entire system is tested for any faults and failures.

**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** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment. (kenny)

Some of the major advantages of the Waterfall Model are as follows −

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are well documented.

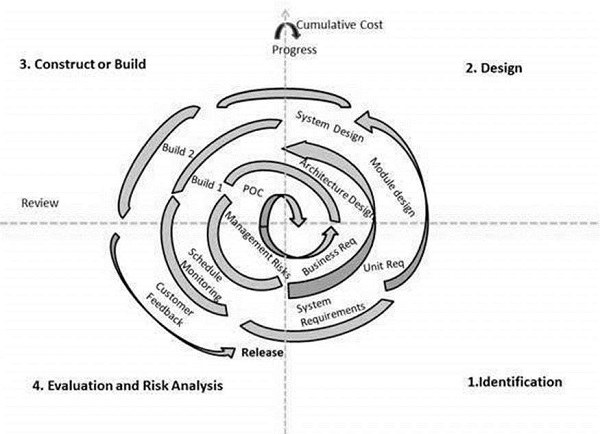
The major disadvantages of the Waterfall Model are as follows −

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.
* Adjusting scope during the life cycle can end a project.
* Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

**SPIRAL**

The spiral model combines the idea of iterative development 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. The spiral model, also known as the spiral lifecycle model, is a systems development lifecycle ([SDLC](http://searchnetworking.techtarget.com/definition/SDLC)) model used in information technology (IT). (kenny)

**Diagram:**

(tutorialspoint)

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

**Identification**

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

**Design**

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

**Construct or Build**

The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

**Evaluation and Risk Analysis**

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback. (James)

The advantages of the Spiral SDLC Model are as follows −

* Changing requirements can be accommodated.
* Allows extensive use of prototypes.
* Requirements can be captured more accurately.
* Users see the system early.
* Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

The disadvantages of the Spiral SDLC Model are as follows −

* Management is more complex.
* End of the project may not be known early.
* Not suitable for small or low risk projects and could be expensive for small projects.
* Process is complex
* Spiral may go on indefinitely.
* Large number of intermediate stages requires excessive documentation.

**COMPARISION OF WATERFALL AND SPIRAL MODEL**

|  |  |
| --- | --- |
| **SPIRAL MODEL** | **WATERFALL MODEL** |
| 1] Spiral model is not suitable for small projects. | 1] Waterfall model is suitable for small projects. |
| 2] Better risk management. | 2] High amount of risk and uncertainty. |
| 3] Process is complex. | 3] Easy to understand. |
| 4] The process may go indefinitely. | 4] Stages are clearly defined. |
| 5] This model is suitable for long and ongoing projects. | 5] This model is not suitable for long and ongoing projects. |
| 6] Iterations are followed | 6] Sequence is followed |
| 7] Flexible with user requirements | 7] Requirements once fixed cannot be modified |
| 8] Refinements are easily possible | 8] Refinements are not so easy |
| 9] Phases  are repeated itself | 9] Phases are processed and completed one at a time. |

**CONCLUSION:**  
From the above description of waterfall and spiral model, the spiral model is an advanced version of waterfall model. Spiral model overlaps the tasks rather than completing the task and starting other while waterfall model completes the task or one step and start other steps.

**REFERENCES:**

# Bibliography

Acodez. (n.d.). *https://www.google.com.np/search?q=waterfallmodel&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjWwK28jMnUAhUST48KHQ5rBisQ\_AUICigB&biw=1360&bih=662#imgrc=bDH279St-cW2tM:*. Retrieved 06 19, 2017, from www.google.com.np: https://www.google.com.np

*https://www.tutorialspoint.com/sdlc/sdlc\_spiral\_model.htm*. (n.d.). Retrieved 06 19, 2017, from tutorialspoint.com: https://www.tutorialspoint.com/sdlc/sdlc\_spiral\_model.htm

*https://www.tutorialspoint.com/sdlc/sdlc\_spiral\_model.htm*. (n.d.). Retrieved 06 19, 2017, from www.tutorialspoint.com: https://www.tutorialspoint.com

James. (n.d.). *http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/*. Retrieved 06 19, 2017, from istqbexamcertification.com: http://istqbexamcertification.com

kenny. (n.d.). *https://www.tutorialspoint.com/sdlc/sdlc\_spiral\_model.htm*. Retrieved 06 19, 2017, from www.tutorialspoint.com: https://www.tutorialspoint.com

tutorG. (n.d.). *https://www.ijcsi.org/papers/IJCSI-12-1-1-106-111.pdf*. Retrieved 06 19, 2017, from www.ijcsi.org: https://www.ijcsi.org

tutorialspoint. (n.d.). *https://www.tutorialspoint.com/sdlc/sdlc\_spiral\_model.htm*. Retrieved 06 19, 2017, from www.tutorialspoint.com: https://www.tutorialspoint.com

unknown. (2014). *http://searchsoftwarequality.techtarget.com/definition/waterfall-model*. Retrieved 06 19, 2017, from searchsoftwarequality.techtarget.com: http://searchsoftwarequality.techtarget.com/definition/waterfall-model

unknown. (n.d.). *http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/*. Retrieved 06 19, 2017, from istqbexamcertification.com: http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/