**ST.XAVIER’S COLLEGE**

**MAITIGHAR, KATHMANDU**

**Software Engineering**

**THEORY ASSIGNMENT-3**

**Submitted by:**

Name: Atul Nepal

Class: BIM 6th Semester

Section: ‘A’

Roll no: 008

**Submitted to**

----------------------------------

Mr. Bal Krishna Subedi

Lecturer

Department of Computer Science

St. Xavier’s College, Maitighar

**Date: 12th June 2017**

# Software

Software, in its most general sense, is a set of instructions or programs instructing a computer to do specific tasks. Software is a generic term used to describe computer programs. Scripts, applications, programs and a set of instructions are the terms often used to describe software. (Software)

# Software Engineering

Software engineering is the application of principles used in the field of engineering, which usually deals with physical systems, to the design, development, testing, deployment and management of software systems. (Software Engineering)

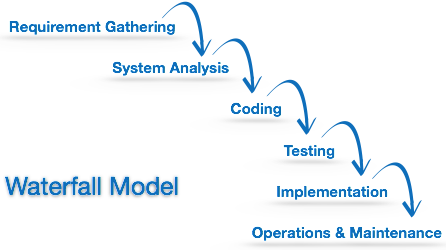
# Software Process Model

A software process model is a simplified representation of a software process. Each model represents a process from a specific perspective.

These generic models are abstractions of the process that can be used to explain different approaches to the software development. They can be adapted and extended to create more specific processes. (Software Engineering — Software Process and Software Process Models (Part 2))

# Waterfall Model

Waterfall model is the simplest model of software development paradigm. It says the all the phases of SDLC will function one after another in linear manner. That is, when the first phase is finished then only the second phase will start and so on.

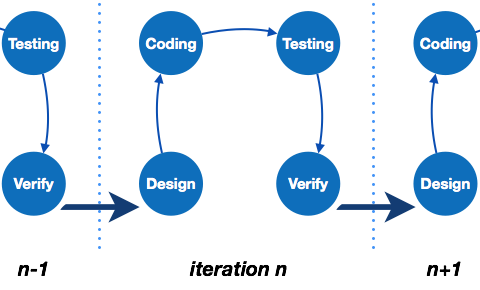


This model assumes that everything is carried out and taken place perfectly as planned in the previous stage and there is no need to think about the past issues that may arise in the next phase. This model does not work smoothly if there are some issues left at the previous step. The sequential nature of model does not allow us go back and undo or redo our actions.

This model is best suited when developers already have designed and developed similar software in the past and is aware of all its domains.

# Iterative Model

This model leads the software development process in iterations. It projects the process of development in cyclic manner repeating every step after every cycle of SDLC process.

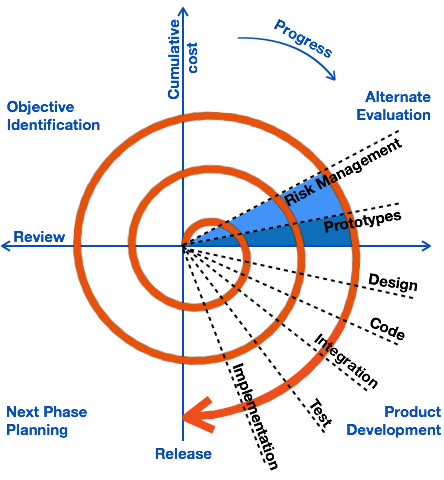


The software is first developed on very small scale and all the steps are followed which are taken into consideration. Then, on every next iteration, more features and modules are designed, coded, tested and added to the software. Every cycle produces a software, which is complete in itself and has more features and capabilities than that of the previous one.

After each iteration, the management team can do work on risk management and prepare for the next iteration. Because a cycle includes small portion of whole software process, it is easier to manage the development process but it consumes more resources.

# Spiral Model

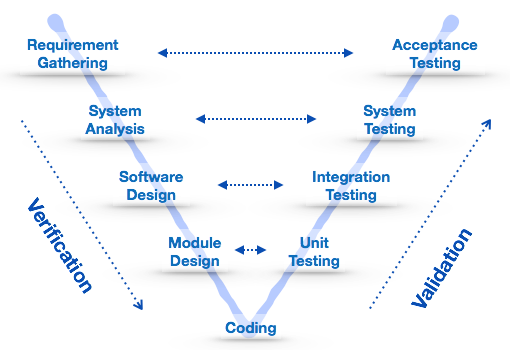
Spiral model is a combination of both, iterative model and one of the SDLC model. It can be seen as if you choose one SDLC model and combine it with cyclic process (iterative model).



This model considers risk, which often goes un-noticed by most other models. The model starts with determining objectives and constraints of the software at the start of one iteration. Next phase is of prototyping the software. This includes risk analysis. Then one standard SDLC

# V – model

The major drawback of waterfall model is we move to the next stage only when the previous one is finished and there was no chance to go back if something is found wrong in later stages. V-Model provides means of testing of software at each stage in reverse manner.

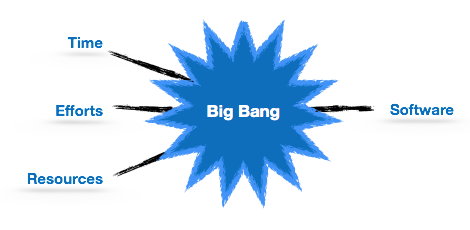


At every stage, test plans and test cases are created to verify and validate the product according to the requirement of that stage. For example, in requirement gathering stage the test team prepares all the test cases in correspondence to the requirements. Later, when the product is developed and is ready for testing, test cases of this stage verify the software against its validity towards requirements at this stage.

This makes both verification and validation go in parallel. This model is also known as verification and validation model.

# Big Bang Model

This model is the simplest model in its form. It requires little planning, lots of programming and lots of funds. This model is conceptualized around the big bang of universe. As scientists say that after big bang lots of galaxies, planets and stars evolved just as an event. Likewise, if we put together lots of programming and funds, you may achieve the best software product.



For this model, very small amount of planning is required. It does not follow any process, or at times the customer is not sure about the requirements and future needs. So the input requirements are arbitrary.

This model is not suitable for large software projects but good one for learning and experimenting. (Software Development Life Cycle)

# References

*Software*. (n.d.). Retrieved June 12, 2017, from Technopedia: https://www.techopedia.com/definition/4356/software

*Software Development Life Cycle*. (n.d.). Retrieved June 12, 2017, from Tutorialspoint: https://www.tutorialspoint.com/software\_engineering/software\_development\_life\_cycle.htm

*Software Engineering*. (n.d.). Retrieved June 12, 2017, from WhatIs.com: http://whatis.techtarget.com/definition/software-engineering

*Software Engineering — Software Process and Software Process Models (Part 2)*. (n.d.). Retrieved June 12, 2017, from Medium: https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc