GENIUS | Page No.
Date: / /

Unit - I Software Engineering & Models

(#) Software Engineering

- Software is a program or set of program containing instructions that provide desired functionality. Engineering is the process of designing and building something that serves a particular purpose.
- Software Engineering is the process of designing, developing, testing and maintaining software.
- It is rapidly evolving field, and new tools and technologies are constantly being developed to improve the software development process.
- Software Engineering is mainly used for large project based on software system rather than single program or application.

(#) Evolving Role of Software

The software evolution process includes fundamental activities of change analysis, release, planning, system implementation and releasing a system to customers.

(i) 1945 - 65 (Origin)

Software engineering was first used in late 1960s during a NATO conference on software engineering.

(ii) 1965 - 85 (Crisis)

The conference was held in response to 'software crisis' - A period characterised by the inability to deliver reliable and efficient software on time.

- The 1970s marked the beginning of the structured programming era
- The 1980s and 1985s saw the rise of object - oriented programming (OOP)

(iii) 1990 - 2000 (Internet)

The 1990s witnessed the rapid growth of internet, leading to software development.

Agile methodologies emerge as a response to the limitation of traditional models like waterfall

(iv) 2000 - 2010 (Light weight)

The era of Open Source and development. Development practise began to emerge, focusing on integration of development.

(v) 2010 - till now (cloud computing and AI)

In the current decade, artificial intelligence and machine learning are increasingly being integrated into software engineering process

Changing Nature of Software

① System Software

Basically system software is a collection of program to provide services to others programs Infrastructure software come under this category like compilers, OS, editors, drivers etc

② Real time Software

These software are used to monitor, control and analyze real world events as they occur.

Ex - Software required for weather forecasting.

③ Embedded Software

This software is placed in "Read - Only - Memory" of the product and control various function of product. The product could be an aircraft, automobile, security system

④ Business Software

This is the largest application area. The software designed to process business applications is called business software

Ex - file monitering, employee management etc.

⑤ Personal Computer Software

The software used in personal computer are covered in this category. Example are word processor, multimedia, animating tools, computer games etc.

⑥ Web based Software

The software related to web application come under this category. Example are CGI, HTML, Java.

Characteristics of Software

① Functionality

Functionality refers to the set of features and capabilities that a software program or system provide to its users.
It includes :-

- ↳ Data storage and processing
- ↳ User Interface and navigation.
- ↳ Communication and networking
- ↳ Security and access control
- ↳ Automation and scripting.

② Reliability

Reliability refers to its ability to perform its intended function correctly and consistently over time. It helps ensure that the software will work correctly and not fail unexpectedly.

③ Efficiency

It refers to the ability of the software to use system resources in the most effective and efficient manner

④ Usability

It refers to the extent to which the software can be used with ease the amount of effort or time required to learn how to use software

⑤ Maintainability

It refers to the ease with which modification can be made in a software system to extend its functionality, improve its performance or correct error.

⑥ Portability

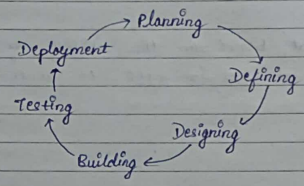
A set of attributes that bears on the ability of software to be transferred from one environment to another without minimum changes.

SDLC (Software Development lifecycle)

- SDLC is cost - effective and time - efficient process that development team use to design and build - high quality software
- The goal of SDLC is to minimize the project risk through foward planning
- This methadology divides the software development process into task you can assign, complete and measure

* Benefits
  - Efficient estimation, planning and scheduling
  - Improve risk management and cost estimation
  - Better customer satisfaction

* Stages of SDLC



The SDLC involves six phrases or stages while developing any software :-

Stage-1) Planning and Requirement Analysis

Planning is a crucial step in everything just as in software development. Requirement analysis is also performed by the developers of the organization. Thus in this stage basic project is designed with all the available information

Stage - 2) Defining Requirement

In this stage, all the requirement for target software are specified. These requirement get approval from customer, market analyst and stakeholders this is fulfilled by utilizing SRS (Software Requirement Specification)

Stage - 3) Designing Architecture

SRS is a reference for software designer to come up with best architecture for the software so that the most practical and logical design is chosen for development

Step-4) Developing Product

At this stage, the fundamental development of product starts for this developers use a specific programming code as per design like C/C++, Python, Java etc.

Stage - 5) Product testing and Integration

After the development of product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC.

At this stage all the probable flaws are tracked, fixed and retested.

Stage - 6) Deployment and Maintenance

After detailed testing, the conclusive product is released in phases as per the organization stategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance.

Advantages

- ↳ Structured Approach
- ↳ Risk management
- ↳ Consistency
- ↳ Collaboration
- ↳ Cost - Effective

Disadvantages

- ↳ Time consuming
- ↳ High upfront cost
- ↳ Inflexibility

Software Development Models

- Software modeling is the process of creating abstract representation of software system.
- These model serve as blueprint that guide developers, designer through the system structure, behaviour and functionality.

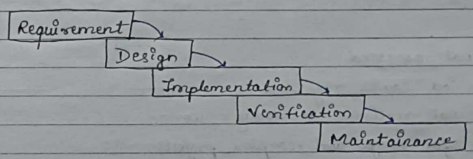
Types

1. Waterfall Model

Waterfall model is a famous and good version of SDLC for software engineering.

Waterfall model is linear and sequential model which means that development phase cannot begin until previous phase is completed.

* Phases of Waterfall Model



↳ Requirement
Requirement of system are collected and documented

↳ Design
Based on the fact how software will built

↳ Implementation
Hardware, software and application programs are implemented

↳ Verification
Software is verified

↳ Maintenance
Problems that arises have to be solved time to time.

Advantage
- Simple and Easy to understand
- Useful for small project
- Easy to manage
- Provide structure way to do things

Disadvantage
- Risk is not assessed
- Testing period comes very late
- Model is not realistic in today's world
- Not good for large and complex projects

② V- Model

V- model is also known as verification and validation model it is widely used in software development process.

Execution of each process is sequential that is the new phase starts only after the previous phase ends.

In this model, Verification phase will be on one side, Validation phase will be on other side that is both the activities run simultaneously and both are connected to each other in V- shaped.

* Phases of V- Model

Requirement analysis — Acceptance testing
System design — System testing
Architecture design — Integration testing
Module design — Unit testing
Coding

- In V- design the left side represent development activities, the right side represent the testing activity.

Advantage
- Simple and easy to use model
- Planning, testing and designing can be done before coding.
- Defect are detected in its intial stage.

Disadvantage

- Not suitable for complex project
- Remains both high risk and uncertainity.
- Not suitable for ongoing project

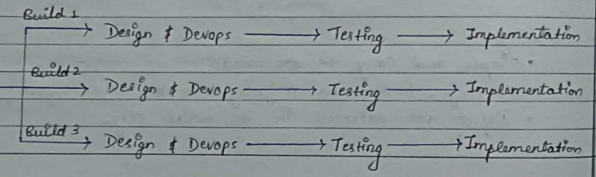
③ Incremental Model

In Incremental model, the software development process is divided into several increment and the same phase are followed in each increment.

Under this model a complex project is developed in many modules or builds.

- It collect the customer's requirement, now instead of making the entire software at once, we first take some requirement and based on them create a module or function of the software and deliver it to the customer. then take some more requirement and based on them add another module to that software.

Similarly module are added to the software in each increment until the complete system is created.



④ RAD Model

RAD Model stand for Rapid Application Development Model. The methadology of RAD model is similar to that of incremental or waterfall model. It is used for small project.

- If the project is large then it is divided into many small project and these small project are planned one by one and completed. By completing small project, the larger project gets ready quickly.

* Phases of RAD Model

Module

Business Modeling
↓
Data Modeling
↓
Process Modeling
↓
Application & turnover
↓
Testing & turnover

↳ Business Modeling

In this phase, the business model is designed on the basis of whatever function the business has.

↳ Data Modeling

Using business model, we had prepared the data object required for business

↳ Process Modeling

The data object that we define in data modeling are converted to establish the business information flow.

↳ Application generation

In this phase we start building the software based on the output of above three phases.

↳ Testing And turnover

Whatever prototype we have made or whatever interface we have are tested in this phase.

Advantage

- Reduces time in development
- Component are reused
- It is flexible we can make changes in it

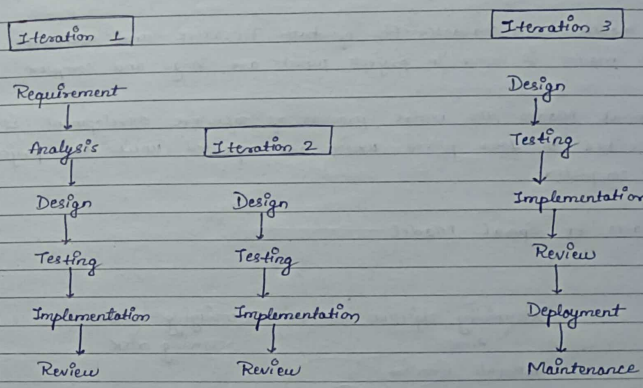
Disadvantage

- Need highly skilled developers and designer
- Difficult to manage
- Not suitable for project that are complex

(5) Iterative Model

In Iterative Model we start developing the software with some requirement and when it is developed, it is reviewed. If there are requirement for changes in it then we develop a new version of the software based on those requirement this process repeats until we get our final product.

* Phases of Iterative Model



Advantage

- Bugs and errors can be identified easily.
- Software prepare quickly
- Testing and debugging become easier.

Disadvantage

- Not suitable for small project
- Requires more resources
- Takes long time to make software

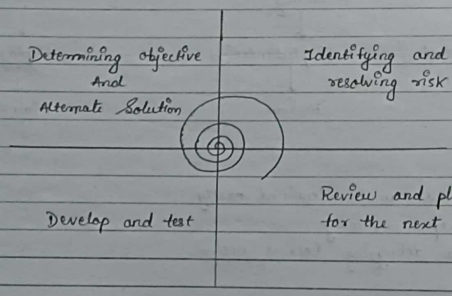
⑥ Spiral Model

Spiral Model is also known as Meta Model. It was first developed by Barry Bohem in 1986.

This model has characteristics of both iterative and waterfall model this model is used in project which are large and complex.

In Spiral Model the entire process of software development is described in four phase which are repeated until the project is completed.

* Phases of Spiral Model



↳ Determining Objective And Alternate Solution

Whatever requirement the customer has related to the software are collected on the basis of which objectives are identified and analyzed

↳ Identifying and resolving risk

All the proposed solution are assessed and best solution is selected

↳ Develop and test

Now the development of software is started and then verified through testing.

↳ Review and plan for next phase

In this phase the developed version of the software is given to the customer and he evaluates it.

Advantage

- Suitable for large and complex project
- Easy to estimate the cost of project
- Risk analysis is done in each phase

Disadvantage

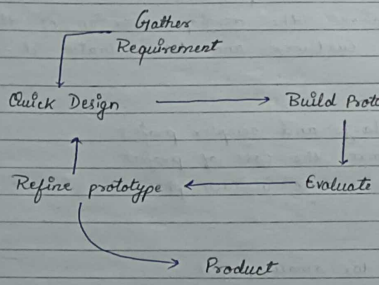
- Not Suitable for small project
- Cost is high
- Requires more documentation than other models.

⑦ Prototype Model

- In this model, first a prototype is created and then the final product is manufactured based on that prototype
- The prototype model was developed to overcome the shortcoming of the waterfall model

Problem in this model is that if the user are not satisfied with the prototype model, then new prototype model is created again due to which this model consume a lot of money and time.

* Phases of prototype model



↳ Requirement gathering
The first step of prototype model is to collect the requirement.

↳ Build the Initial prototype
In this phase the initial prototype is built, in this some basic requirement are displayed.

↳ Review the prototype
When the construction of the prototype is completed, it is presented to the user and feedback is taken from them.

↳ Revise and improve the prototype
When feedback is taken from the user, the prototype is improved on the basis of feedback.

Advantage
- Easy to detect error
- flexible in design
- we can find missing functionality easy.

Disadvantage
- This model is costly
- May increase the complexity of system

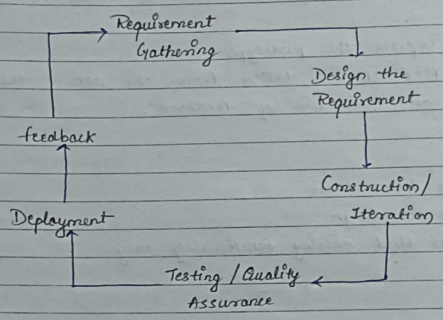
⑧ Agile Model

Agile model is the combination of iterative and incremental model, that it is made up of iterative and incremental model.

The agile model was created mainly to make changes in the middle of software development so that the software project can be completed quickly.

In Agile model the software product is divided into small incremental part, smallest part is developed first and then the larger one.
And each incremental part is developed over iteration.

* Phases of Agile Model



↳ Requirement gathering
In this phase the development team must gather the requirement by interation with the customer.

↳ Design the Requirement
In this phase the development team will use user - flow diagram to show the working of the new features.

↳ Construction / Iteration
In this phase, development team member start working on their project.

↳ Testing / Quality Assurance
In this step development team member test their project which was constructed or iterated

↳ Deployment
In this step development team will deploy the working project to end user.

↳ Feedback
This is the last step of agile model, In this the team receives the feedback about the product and works on correcting the bugs or errors

Advantage
- Reduces the development time of whole project
- focus on customer value
- Delivery high quality software

Disadvantage
- Lack of predictibility
- Increased Overhead
- Depends on team