

CH.1 : Software Development Process Models

WHAT IS SOFTWARE ?

Definition :

Software is collection or set of computer programs, procedures, rules or associated documents

it includes audio, video, images etc.

WHAT IS SOFTWARE ENGINEERING ?

Definition :

1. Software engineering is an engineering approach that covers various stages from planning to maintenance stage
2. SE is an engineering discipline that delivers high quality s/w at agreed cost & in planned schedule.

SOFTWARE APPLICATION DOMAIN

Area of software that it can operate easily

TYPES OF SOFTWARE

mainly 2 types : Application Software

System Software

APPLICATION SOFTWARE :

- > User requirement
- > Designed to fulfil user requirements
- > Ex., MS Excel, Browser

SYSTEM SOFTWARE :

- > interface b/w software & hardware
- > Designed to manage control of hardware and software

--> Ex., OS, Drivers

CHARACTERISTICS OF GOOD SOFTWARE :

- 1.Maintainability
- 2.Budget friendly
- 3.Reliability
- 4.Readability
- 5.Reusability
- 6.Modularity
- 7.Functionality
- 8.Portability
- 9.Proper Documentation
- 10.Correctness

SPECIAL CHARACTERISTICS OF SOFTWARE :

1.Software doesn't wear out

any hardware after long use it can be damaged but in case of software it can't be damages , through maintenance it can be updatable

it can't be affected by "environmental effects"

2. Software is engineered not manufactured

Hardware can't be modifiable but software can modifiable for use later on

3. Reusability of component

Once software created it should be scalable , it should have ability to reuse its components

4. Software is flexible for custom built

Software is manufactured from gathering user requirements and for fulfil requirements

PROGRAM V/S SOFTWARE

1. small in size	1.large in size
2. single developer	2. team of developer
3. may not be structured	3. well structured
4. may not be scalable	4. can be scalable
5. no or less documentation	5. well documented
6. may not be maintainable	6. maintainable

SOFTWARE ENGINEERING LAYERED APPROACH :

Four Layers

1. Quality

it is first layer of layered approach

software mainly focuses on quality

it checks whether output satisfies user's requirement or not

every organization should focus on its software quality

2. Process

it is second layer of layered approach

it is heart of software engineering

main idea behind add process in layered approach is to delivery good software in timely manner

3. Method

it is third layer of layered approach

it describes 'how to build' and in which method software should be developed

4. Tools

it is fourth and last layer of layered approach

it executes process in proper layer

it supports of its above layer

WHY SOFTWARE ENGINEERING ?

1. To develop high quality software product
2. To develop large problems
3. To achieve skill of problem solving
4. To construct project with agreed cost
5. To deliver product with timely manner
6. To become good programmer / developer

UMBRELLA ACTIVITIES

TRICK : "Please Follow Simple Rules During Real Software"

1. P – Project Tracking & Control
2. F – Formal Technical Review
3. S – Software Quality Assurance (SQA)
4. R – Risk Management
5. D – Document Preparation & Production
6. R – Reusability Management
7. S – Software Configuration Management (SCM)

GENERIC PROCESS FRAMEWORK

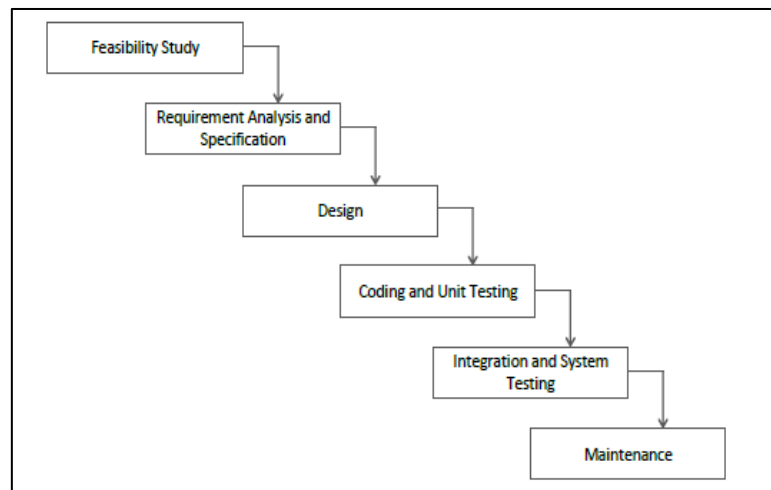
1. Communication
2. Planning
3. Modelling
4. Construct
5. Deployment

SOFTWARE DEVELOPMENT LIFE CYCLE

1. Birth/Dead
2. Development
3. Maturity
4. Maintenance

1. Planning
2. Analysis (SRS,Size estimation etc)
3. Designing
4. Coding/ Implementation
5. Testing
6. Maintenance

CLASSICAL WATERFALL MODEL



1. Feasibility Study

analysis of project whether it is feasible or not

feasible means budget, team management, time duration etc.

2. Requirement Analysis & Specification

SRS

3. Design

Different diagrams of software

4. Coding & Unit Testing

Actual Implementation of product

5. System Testing & Integration

Combine small units

Various type of testing like black box testing, Whitebox testing, alpha, beta testing etc...

6. Maintenance

Maintain the product

maximum effort (60%)

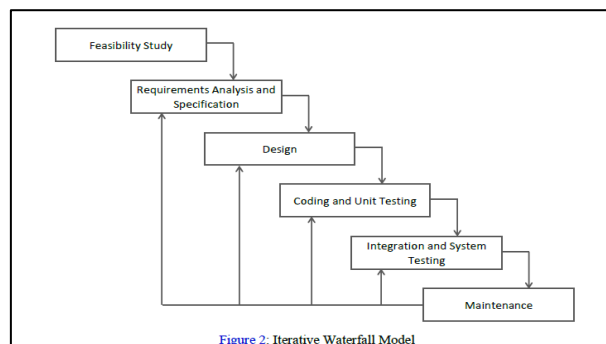
Advantages :

Base Model, Easy to Implement, Simple, Small Project

Disadvantages :

No Feedback, High-risk, 60% effort in maintenance, No Experiment, No Parallelism (Team can't work parallelly)

ITERATIVE WATERFALL MODEL



Modified version of classical waterfall model

Difference b/w classical & iterative is feedback

No feedback in Feasibility study

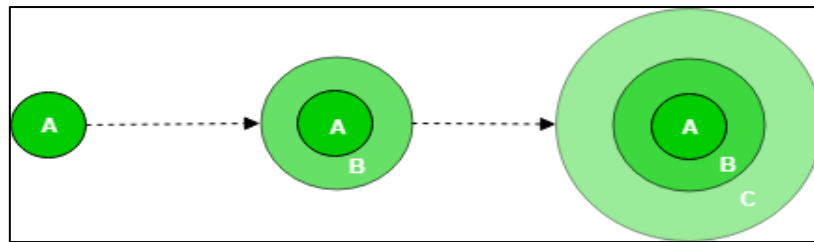
Advantages :

Base Model, Easy to Implement, Simple, Small Project, Feedback

Disadvantages :

High-risk, 60% effort in maintenance, No Experiment, No Parallelism (Team can't work parallelly), Not Efficient, Less customer interaction

INCREMENTAL MODEL



Specially for large project

module by module development

ex., College : Faculties, Students, HODs, Attendance are the modules of College management system

1. Requirements (SRS)

2. Design&Delopment --> Testing --> Maintenance //for each & every model

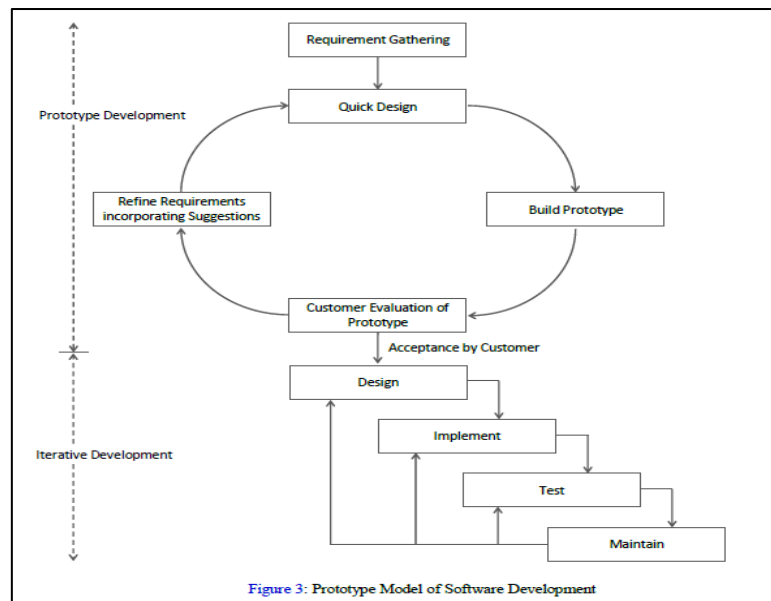
Advantages :

MaximumCustomerInteraction, ForLargeProducts,
BenificialForEarlyReleaseDemand, FlexibleToChange

Disadvantages :

NeedsGoodPlanning, SometimesComplexToDivideModules

PROTOTYPE MODEL



Make replica before actual product

User can give feedback

If user's requirements are not clear then this model is worst

1. Requirement Gathering

2. Quick design

3. Refine Requirements

4. Customer Evolution

if customer accepts then

4.1. Design

4.2. Implement

4.3. Test

4.4. Maintain

if no then repeat 2,3....

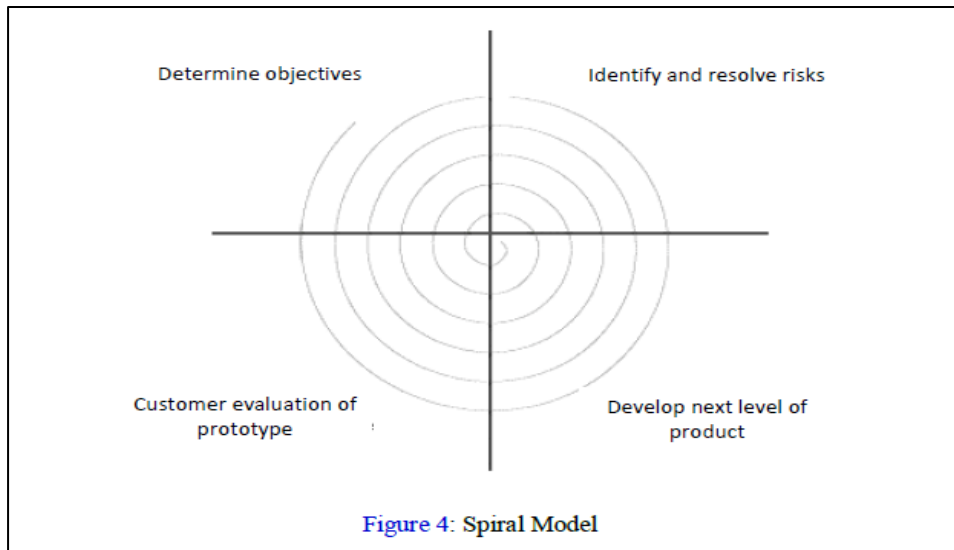
Advantages :

GoodForTechnicalAndRequirementRisks,
MoreChancesOfUserSatisfaction, FlexibleForDesign&Development,

Disadvantages :

Costly For Development, Sometimes Wasting Of Time,
if user not satisfied with initial prototype then it can be lost interaction

SPIRAL MODEL



Best for risk management

1. Objective determination & identify alternative solutions
2. Identify & resolve risks
3. Develop next version of product
4. Review & plan for next phase

if user not satisfied then it continues like loops

for large products

used when risk is in every phase

radius of spiral is equal to cost

angular dimension shows progress(if 180 then half complete,360 then one spiral is completed likewise...)

Meta model because it uses multimodal like.. Iterative , prototype

Advantages :

Risk Handling, For Large Products, Flexible, More Customer Satisfaction

Disadvantages :

Complex, Costly, Much More Risk Analysis, Time-consuming

AGILE DEVELOPMENT MODEL

Latest model in compare of non-agile models

quick & Efficient

for large models

1. Divide product into small chunks (Iteration)
2. Release
3. Feedback
4. Enhance
5. Re-release

Types :

- Scrum
- XP

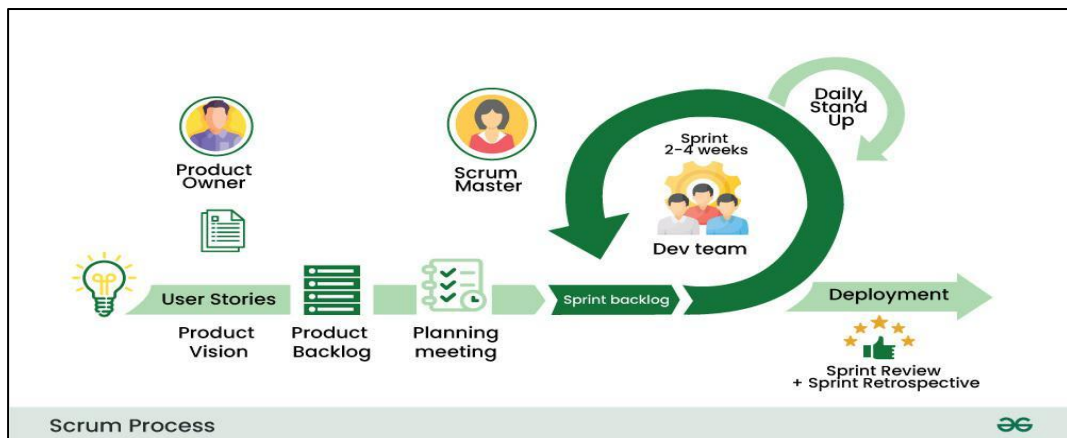
Advantages :

Frequent Delivery, Face To Face Communication, Less Time Consuming

Disadvantages :

Less Documentation, Regular Maintenance Required

SCRUM MODEL



Most popular agile methodology

Lightweight (not complex), iterative (step by step), incremental framework

Scrum breakdowns the development phases into stages or cycles called as **"SPIRINTS"**

Time boundary is here

One sprint at a time

Team has scrum master and product owner with constant communication on daily basis

Widely Used

Daily Scrum : daily meeting of duration 10 to 15 minutes

Backlog : Form where all details of product is mentioned

Scrum master : monitor of team that communicates with client

Product Owner : Client

Advantages :

Freedom, High-quality, LowRiskProduct, LessTimeConsuming, Customer Satisfaction, Reviewing

Disadvantages :

MoreEfficientForSmallTeam (not for large teams), NoChangesInTheSprint

eXTREME PROGRAMMING (XP) MODEL

Suggest quick release

bugs can be detect easily

can be monitoring that how product works

Advantages :

Well Performance, LessDocumentationNeeded

Disadvantages :

Time consuming, High Cost