

## To be use for extra act

### Introduction

- └ Intro to SE
- └ imp of SE
- └ evolution of SE
- └ Characteristics of SE
- └ Software component
  - └ software applications
  - └ software crisis
  - └ Software engineering problems

### SDLC Models

- └ waterfall model
  - └ Incremental Model
  - └ RAD
  - └ prototyping
  - └ spiral model
- } compare  
adv.  
disadv.

# INDEX

NAME: \_\_\_\_\_ STD.: \_\_\_\_\_ SEC.: \_\_\_\_\_

SCHOOL/COLLEGE: \_\_\_\_\_ Roll No.: \_\_\_\_\_

E-mail: \_\_\_\_\_ Mobile: \_\_\_\_\_

S.No.	Date	Title	Page No.	Remarks
		Software as a product		
		When software is developed for a specific requirement then it is known as software product.		
		General		
		Generic - sold in open market eg. MS office		
		Customized - developed for a specific customer (Bespoke)		eg. airline traffic control
		Software products are produced using software engineering		
		Software applications		
		GPS, Instagram, whatsapp, Traffic lights, fire alarms, automobiles, home and office automation, security systems.		
		Collecting customer's requirements		
		Preparing the design		Phases
		Writing programs or code		
		Testing		
		Maintenance		

S.No.	Date	Title	Page No.	Remarks
		In order to create a complex s/w we should use SE technique as well as to reduce the complexity we should use abstraction and decomposition.		
		packy documentation		
		KB - program - set of instructions used to perform a specific task		
		MB / unorganized & limited functionality & few features		
		unplanned & procedural		
		A program may not be software but a software can be a program		
		<ul style="list-style-type: none"> <li>- takes more time to build</li> <li>- MB to GB</li> <li>- comprehensive documentation, user manual.</li> <li>- planned, organized and systematic</li> </ul>		
		Complex to create } diff to understand } expensive to change } Challenging task. } unreliable, expensive to maintain & use		

Software → programs, ds and documentation (data file) of Chrome, VLC Movie player

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

### Software Engineering

Software engineering is the product of two words software and engineering.

Software is a collection of integrated processes.

SE: It is systematic, disciplined, cost effective techniques for software development operation & maintenance

Engineering approach to develop a software using well defined scientific principle method & procedure

SE: is required

to manage large software  
for more scalability,

cost management

to manage the dynamic nature of software  
for better quality management

SE is a process in which user needs are analysed and s/w is designed based on their needs

SE is a technique through which we can develop or create a software for computer systems & any other electronic devices

A detailed study of engineering to the design, development and maintenance of software is known as SE

### SDLC

A software life cycle is a pictorial and diagrammatic representation of the software life cycle. SDLC is a part of different processes. It has different phases. SDLC is divided into different SDLC models.

The term software engineering was first proposed in 1968 at a NATO conference held to discuss what was then called as software crisis.

1965 - 1985 → software crisis period

Software was unreliable, cost more than expected & was delivered late.

OS 360 - eg of software crisis

SE techniques & methods were developed such as structured programming, info hiding & object-oriented development.

Tools and standard notations were developed and are now extensively used.

- Internet, e-commerce, banking ← done by SE

### software terminologies

#### 1) Deliverables

Deliverables are generated during the software development for eg. source code, user manuals, operating procedures and other required documents.

#### 2) Milestones

After the deliverables, the events that are used to identify the status of the software.

#### 3) Finalizing the specific task

design documentation completion.  
completion of project planning & management

#### 4) Product

A delivered part which includes source code specification document, manuals is called product.

#### 5) A process

It is the way in which a software or a product developed. It is a collection of activities which are required for good quality software.

### 6) Measurement

A measure is a quantitative term which indicates dimensions size capacity, efficiency, productivity.

Measurement is a evaluation of measure.

Metric is a quantitative measure of degree for a particular process or attributes.

### 7) Productivity

It is define as a ratio of output to per unit of efforts.

## Characteristics of a good software engineer Importance of Software Engineering

### 1) Reduce complexity

SE divides big problems into various small issues. And then start solving each small issue one by one. All these problems are solved independently to each other.

### 2) To minimize software cost

In SE programmers project everything and decrease all those things that are not needed. In turn the cost for software production becomes less as compared to any software that does not use SE method.

### 3) To decrease time

SE is <sup>not</sup> a very time consuming procedure. If you are making your software acc. to the software engineering method, then it will decrease a lot of time.

### 4) Reliable software

In SE testing and maintenance are given, so there is no worry of its reliability.

Date. \_\_\_\_\_  
Page No. \_\_\_\_\_

Date. \_\_\_\_\_  
Page No. \_\_\_\_\_

### 5) Effectiveness

Software becomes more effective in cut off with the help of SE

### Purpose of SE

To manage large software  
for greater scalability

To manage the cost

To do better quality management

### Characteristics of good software

Operational

Transitional

Maintenance

### Operational

In operational categories, the factors that decide the software performance in operations. It can be measured on

- Budget
- Usability
- Efficiency
- Correctness
- Functionality
- Dependability
- Security
- Safety

### Transitional

When the software is moved from one platform to another, the factors deciding the software quality

- Portability
- Interoperability
- Reusability
- Adaptability

### Maintenance

In this categories all factors are included that describes about how well a software has the capabilities to maintain itself in the ever changing environment

- Modularity
- Maintainability
- Flexibility
- Scalability

~~we should be able to do changes in the software based on business needs.~~

Software Characteristics are classified into six major components

### Functionality

It refers to the degree of performance of the software against its intended purpose

compliance security Interoperability Suitability accuracy

~~should be reliable  
should not cause damage  
or financial loss  
during failure~~

Date: \_\_\_\_\_  
Page No. \_\_\_\_\_

Date: \_\_\_\_\_  
Page No. \_\_\_\_\_

### Reliability

A set of attributes that bears on the capability of software to maintain its level of performance under the given condition for a stated period of time.

Recoverability      Maturity      Fault tolerance

### Efficiency

~~fast, consume less memory, not waste computer resources, timespan~~  
It refers to the ability of the software to use system resources in the most effective and efficient manner. The software should make effective use of storage space and execute command as per desired living requirements.

In Time      In Resource.

### Usability

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

Learability      Operability      Understandability

### Maintainability

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

Testability      Stability      Changability      Operability

### Portability

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

Adaptability      Installability      Replaceability

### Security

Unauthorized or unknown persons should not be allowed to use the software as they can create harm to the system.

Acceptability: All types of users must be able to use the software. It should be easy to understand, set & install on different systems.

Date: \_\_\_\_\_  
Page No. \_\_\_\_\_

### Software Crisis

Software crisis is a term used in computer science for the difficulty of writing useful and efficient computer programs in the required time.

The software crisis was due to using the same workforce, same methods, same tools even though rapidly increasing in software demand, the complexity of software and software challenges.

If we will use the same workspace, same methods, and same tools after the fast increase in software demand, software complexity and software challenges then there are some problems like software budget problems, software efficiency problem, software quality problems, software managing and delivering problems etc. This condition is called software crisis.

Increasing  
Demand

Increasing  
Complexity

Increasing  
Challenges

### Software Crisis

Same  
workforce

Same  
methods

Same  
Tools

new  
size  
quality  
cost

Date: \_\_\_\_\_  
Page No. \_\_\_\_\_

### Causes of Software Crisis

- cost of buying and maintaining software was as expensive as developing the software.
- At that time projects were running over-time
- software were very inefficient
- quality of software was low
- did not meet user requirements.
- software never delivered
- difficult to alter, debug and enhance
- complexity is hard to change

SE is one possible solution to software crisis.

### Types of Software

System Software: OS, Text Editors, compilers

Programming software: tools for writing computer programs, such as Java, Python, C, C++ etc

Application software: software for a particular task such as accounting, data processing, games etc

embedded software  
web application  
AI software

Scientific Engineering  
Software

Date. \_\_\_\_\_  
Page No. \_\_\_\_\_

Build and fix methods.

No specification, no design, no planning

SDLC

Phases - first page

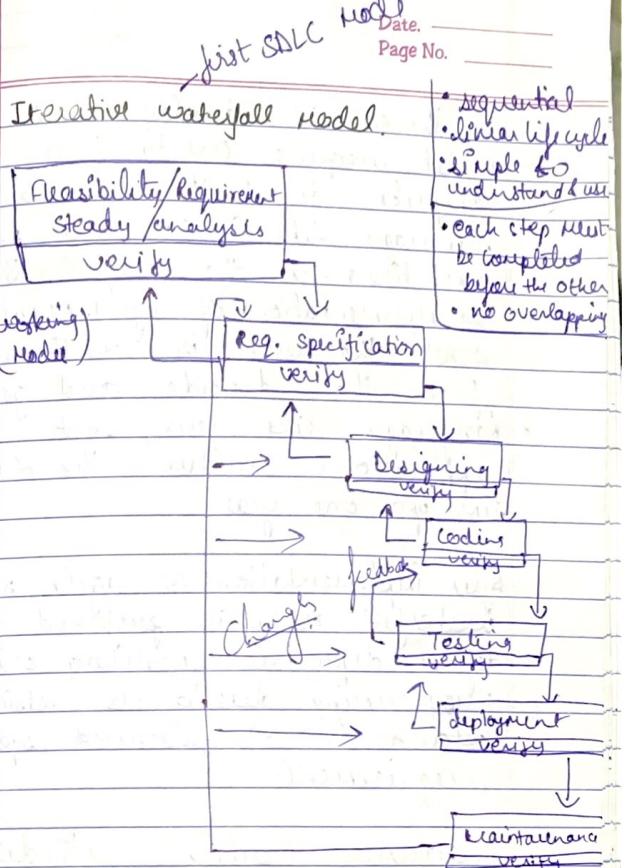
The major software development activities include:

gathering requirements,  
software design  
implementation  
code testing  
documentation  
maintenance

There are various ways in which the above activities can be done to achieve the final software. These are known as software development models.

- simple, easy to understand, well documented  
- per well defined user requirements  
- progress can be tracked accurately  
- schedules development & maintenance cost

It is rigid, not good for complex or long ongoing software systems, no working model in the middle



- It is a linear sequential model which is helpful for banking, accounting, customer, enterprise applications like supply chain management, customer relationship management, inventory management, point of sales, retail management.

Date. \_\_\_\_\_  
Page No. \_\_\_\_\_

Date. \_\_\_\_\_  
Page No. \_\_\_\_\_

- In the earlier development this model require one to two years to complete due to the reason that languages like C, C++ and no competitor in the market. But after the implementation of agile manifesto in 2001 the model is continued from the decade and going on.
- Languages like java, .net, web based applications reduce the development time upto one year

After implementation of agile now the waterfall model is preferred approach defense department, military aircraft programming due to its strict standards and followed specific requirements.

Requirement feasibility study  
There are three types of feasibility study.

Financial  
Technical  
Social

Requirements by the client are documents during this phase and can be clarified by the client later on.

#### Requirement Specification

After fixing all the requirements the development team properly organised and documented it into SRS.  
It is the contract b/w client and the team.

#### System Design

Senior members of the team architect the different levels of projects.

#### High Level design

Low level design  
selecting the approach and language associated with it.

#### Coding

Source code is generated along with the help of programming standards.

#### Testing

Checking all the defects, errors, bugs, faults in the application

- $\alpha$ ,  $\beta$  and acceptance testing is done
- Stress, mutation, smoke, regression are few of the testing which are also implemented during this phase.

### Deployment.

Installation of applications onto the client server.

### Maintenance.

Team ensure that application runs smoothly.

Resolving the issues during ~~built~~ <sup>the</sup> usability phase.

A software development process consists of a no of phases in sequence

A phase can be started only after the completion of previous phase

This development model is sequential  
The phases do not overlap each other

Work can be divided, acc to phase, among diff classes of specialists

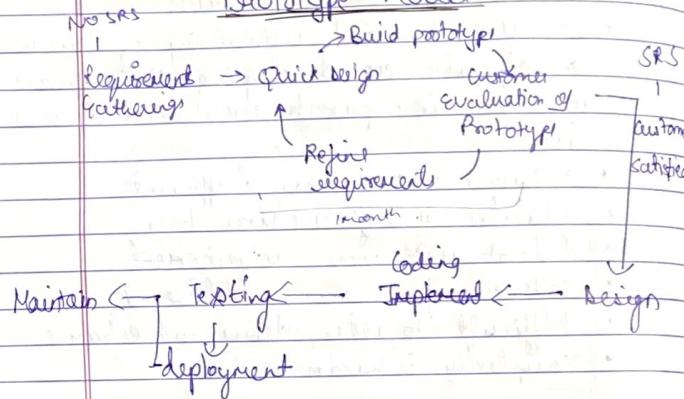
working prototype of the software is given to the user for comments and feedback

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

It helps the user to give all their requirements in a more detailed fashion

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

### Prototype Model



An original model is being developed in the form of prototype/blueprint  
This model might involve extra cost and more time but, the ~~overall~~ <sup>average</sup> cost turned out to be less

Better understanding of customer needs enhances critical examine the technical issues  
It is very useful for large projects  
Easier for quick eliciting of requirements

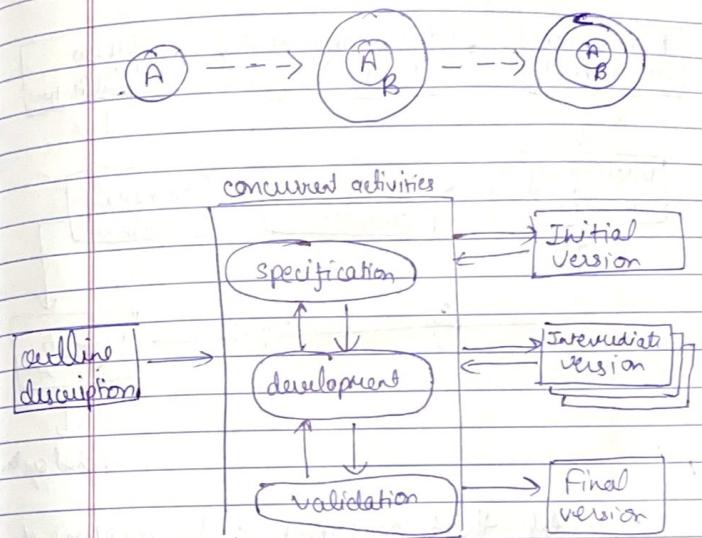
**Advantages**

- reduce maintenance cost
- errors can be detected
- good where requirements are changing.
- reduce the risk of incorrect user requirement

**Disadvantages**

- diff know how the program will cost
- prototyping tools are expensive
- time consuming process
- require extensive customer collaboration

### Evolutionary Model



Evolutionary Model resembles iterative Model but the good thing about this model is that a usable product at the end of each version

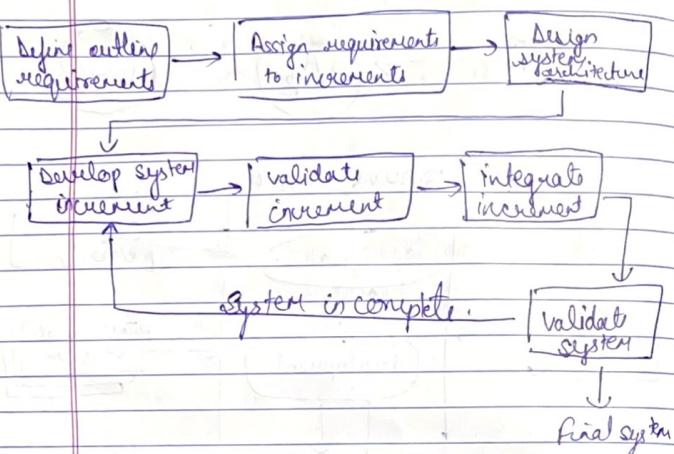
The implementation is categorized according to the user.

This model is very much helpful for complex projects using new technologies and delivery of different functionalities cannot be done in one sheet.

Date. \_\_\_\_\_  
 Page No. \_\_\_\_\_

(Iterative Enhance)

Incremental Model)



- Model that generates diff versions of the software

Software product is developed in increments or steps.

First a few functions are developed to get a working model of the software. Later more functions are added.

Requirement analysis and design are done similarly to the waterfall model.

Coding, integration & testing are done in an incremental way.

Every subsequent release of the module adds function to the previous phase. The process continues until the complete system is achieved.

**Requirement**  
 The product analysis expertise identifies the requirements.  
 This phase performs a crucial role.

**Design & Development**  
 design of the system functionality development methods are finished with success

**Testing**  
 checks the performance of each existing function as well as additional functionality.

**Implementation.**  
 involves final coding and tests the functionality in testing phase.

after this, the no of product working is enhanced and upgraded up to the final system product

errors are easy to recognize if client gets easy to test & debug More flexible functionality early.

cost is high, need good planning - disadvantages

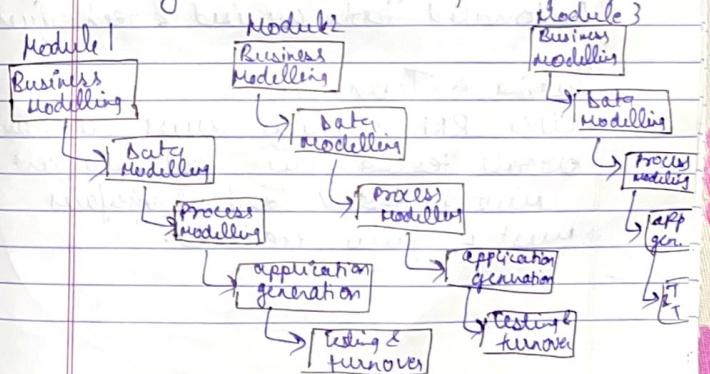
we need this model

- when requirements are superior
- when customers demands a quick release of product
- a project has a lengthy development schedule
- team are not very well skilled or trained

linear sequence  
software developer  
process

### RAD (Rapid Application Development) Model

- It is based on prototyping & iterative development with no specific planning involved.
- The software product is developed in a short period of time (60-90 day)
- This model uses powerful development tools and techniques
- Components or functions are developed in parallel as if they were mini-projects
- Developments are time-boxed, delivered, and then assembled into a working prototype
- can quickly give the customer something to see and use and provide feedback regarding the delivery and their requirements
- Customer requirements are gathered through workshops or focus groups



Date: \_\_\_\_\_  
Page No. \_\_\_\_\_

Date: \_\_\_\_\_  
Page No. \_\_\_\_\_

### Business modelling

Info flows → what data drives,  
what data is generated,  
who generates it,  
where does the info go,  
who processes it and so on

### Data modelling

Data collected is refined into a set  
of data objects (entities). Attributes  
defines the relation b/w these obj.

### Process modelling

Processes are created for adding,  
modifying, deleting or retrieving a  
data object.

### Application generation

Automated tools are used & techniques

### Testing & Turnover

Since RAD emphasis reuse, it reduces  
overall testing time but new part  
must be tested and all interfaces  
must be fully exercised.

adv

flexible for change, changes are adoptable  
reduce development time,  
increases the reusability features

disadv

required skilled people  
cannot use for smaller projects  
on high technical risk it's not suitable  
requires user involvement

### We need this model

- when requirements are well known
- technical risk is limited
- to create project in short span time

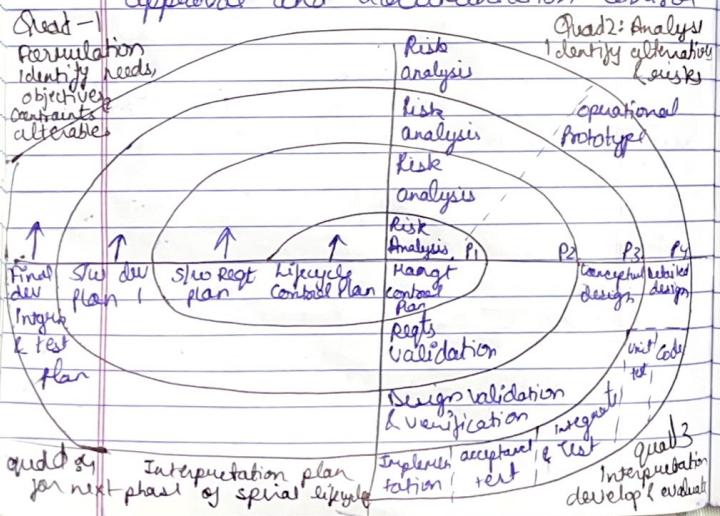
## Spiral Model

It integrates the characteristics of the waterfall model, the incremental implementation and the evolutionary prototyping approach.

Software is produced quickly suitable for high-risk projects

Flexibility in requirements and good for large & complex projects

Good for customer satisfaction & strong approval and documentation control



adv. high amount of risk analysis useful for large and mission critical projects

disadv. can be a costly model risk analysis needed - expertise doesn't work well for smaller projects

we need this model

- when the project is large
- changes may require at any time
- large & high budget project
- requirements are unclear & complex

No fixed phase. Each quadrant of the spiral corresponds to a set of activities for all phases

Formulation

Analysis

Interpretation

Plan next phases

Risk assessment is an imp activity

## Questions

Date. \_\_\_\_\_

Page No. \_\_\_\_\_

Orals

Define software engineering & explain its need

Types of software

Applications of software

SPLC

necessary characteristics of software

Date. \_\_\_\_\_  
Page No. \_\_\_\_\_

- Q Compare waterfall and spiral models
- Q Identify atleast two activities in each quad of a spiral model
- Q Discuss the adv. & disadv. of the prototyping model.
- Q Explain the prob in waterfall model and why it unsuitable for real-world applications.