Page No.
Date

CHAPTER 1

INTRODUCTION TO S.E

① what is a software?

software is :

(a) Instructions that when executed provide desired features, function and performance.

(b) data structure that enables the programs to manipulate information.

(c) descriptive information in both hard copy & virtual forms that describe the operation and the use of the programs.

* Software is an information transformer - producing, managing, acquiring, modifying, displaying or transmitting information that can be as simple as a single bit or as complex as a multimedia ppt.

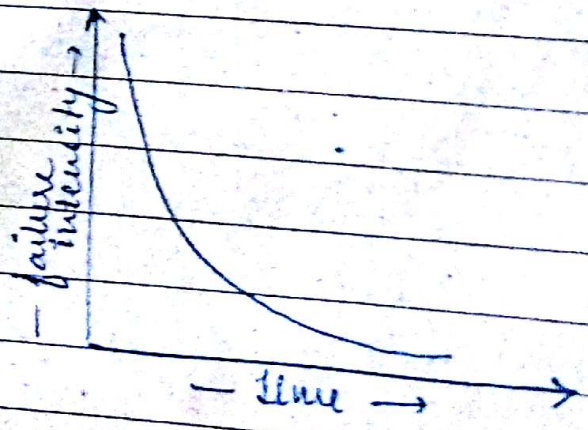
* Software acts as a basis for the control of the computer (OS), the communication of information (networks) and the creation and control of other programs.

Page No.
Date

* Characteristics of Software :

(1) Software is developed or engineered; it is not manufactured in the classical sense.
:- It is one time development effort and continuous maintainance effort in order to keep it operational.

(2) Software doesnot wear out :- Software becomes reliable overtime instead of wearing out. It becomes obsolete, if the environment for which was developed changes.



(3) Softwares are flexible.

(4) Reusability of components : Components of software can be used again in another pro

Page No.
Date

Types of softwares

(1) SYSTEM SOFTWARE – It is a collection of programs to provide service to other programs.
eg: compilers, OS, drivers etc.

(2) APPLICATION SOFTWARE : Any program or group of program designed for end-users.
Eg spreadsheets, web browsers etc.

(3) REAL TIME SOFTWARE : These software are used to monitor, control and analyze real world events as they occur
eg: weather forecasting

(4) BUISNESS SOFTWARE – largest apph area.
→ software designed to process buisness apph.
eg: Payroll, file monoitoring system etc.

(5) Artificial Intelligence Software : - It makes use of non-numerical algorithms to solve complex problems. Eg: expert systems, artificial neural network etc.

(6) Web based software : Software related to web applications. Eg: HTML, Java etc.

(7) Embedded software : These type of software are played in ROM and control various func^n of the products
:- It handles hardware components and is also termed as intelligent software.
eg: security system, automobile, aircraft

Software Myths.

(1) softwares are easy to change : By changing the source code.

(2) Testing software or providing software correct can remove all errors. - Testing can show the presence of errors : Effective test cases finds max. possible errors.

(3) Software can work right at the first time.

(4) Software with more features is a better soft

Page No.
Date

(5) Computers provide greater reliability than the devices they replace.

* Software Crisis : ~~Crisis occured because of~~ inability to develop bug free software.

(1) The Y2K problem was the most crucial problem of last century. It was simply the ignorance about using last 2 digits of the year.

Eg: instead of writing 1964, it was shortened to 2-digit format as 64. The problem arised while writing the year 2000.

(2) Star wars program of USA produced a missile for defence. The Patriot missile failed several times due to a software bug. The error being in the system clock.

(3) "One little bug, one big trash" of Ariane-5 space rocket. The cost of this project was 700 million dollars and within 39 sec of the launch of this rocket, it was destroyed.

Software Process :

* 'PROCESS' is an adaptable approach that enables the people doing the work (the software team) to pick and choose the appropriate set of work actions and tasks. The intent is always to deliver software in a timely manner and with sufficient quality to satisfy those who have sponsored the creation and those who will use it.

Generic process framework for S.E :-

(1) COMMUNICATION – Before any technical work can commence, it is important to communicate and collaborate with the customer.

* The purpose is to understand the objective of the customer and gather requirements that help define software features and function.

(2) PLANNING :- A software project is a complicated journey and the planning creates a map that helps guide the team. The map is called the 'software project plan' which defines the SE work by describing the technical tasks

Page No.
Date

be conducted, the risks that are likely and the resources that will be required, the work products to be produced and a work schedule.

(3) MODELING : A software engineer creates a model to better understand the software requirements and the design that will achieve those requirements.

(4) CONSTRUCTION : This activity combines code generation and the testing that is required to uncover errors in the code.

(5) DEPLOYMENT : The software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

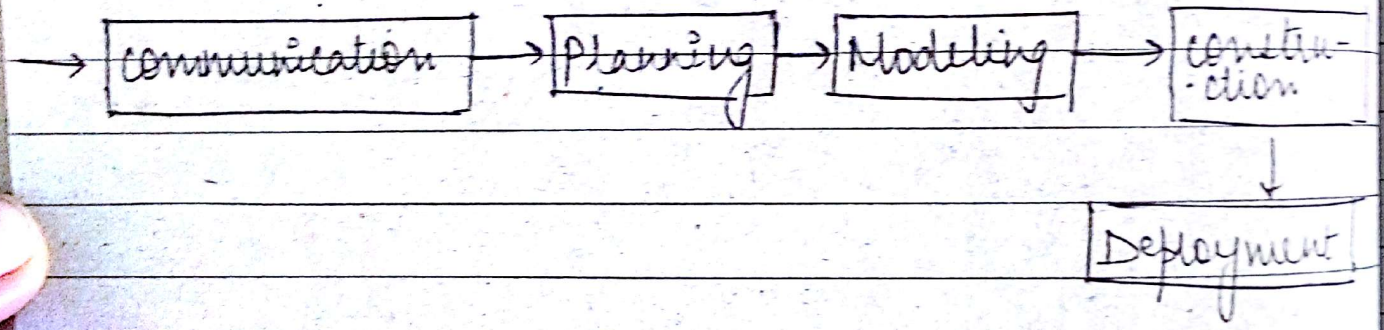
—x—

Page No.
Date

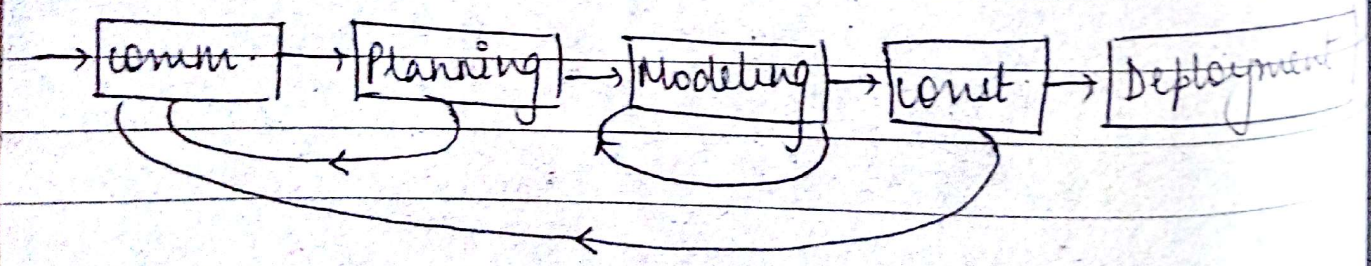
Chapter 2

Process Models

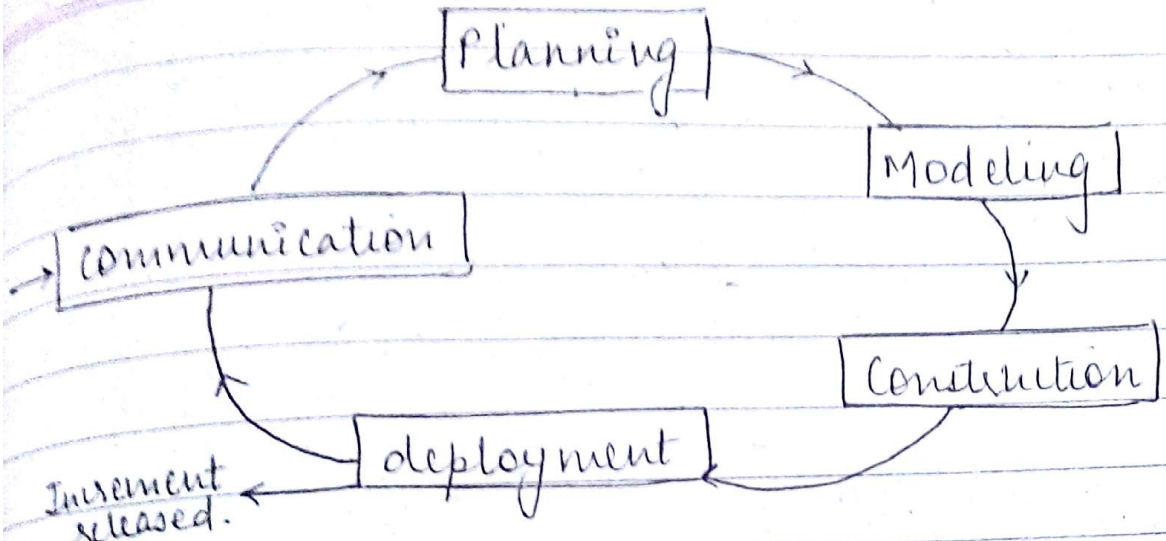
(1) Linear process flow - It executes each of the five framework activities in sequence, beginning with communication and culminating with deployment.



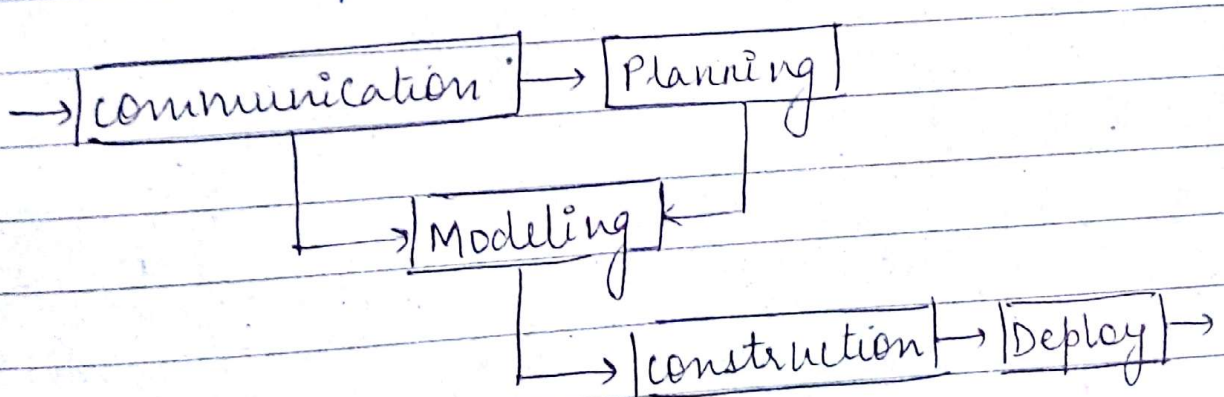
(2) Iterative process flow - It repeats one or more of the activities before proceeding to the next



(3) Evolutionary Process flow - It executes the activities in a circular manner.



(4) Parallel process flow : It executes one or more activities in parallel with other activities.



Capability Maturity Model Integration

* CMMI is a process model that provides a clear definition of what an organisation should do to promote behaviors that lead to improved performance.

* with 5 "maturity levels" or 3 'capability levels', the CMMI defines the most imlpt elements that are required to build great products, or deliver great services and wraps them all up in a comprehensive model.

* CMMI also helps to identify and achieve measurable buisness goals, build better products, keep the customer happier and ensure that we are working as efficiently as possible.

* The output of CMMI project is the suite of products which provides an integrated approach across the enterprise for improving processes, while reducing the redundancy, complexity and cost.

Waterfall Model

→ most familiar model
→ This model has 5 phases :-

① Requirement Analysis and specification phase :
* The goal of this phase is to understand the exact requirements of the customer and to

document them properly.

* The goal is to document all functions, performance and interfacing requirements for the software.
* The requirement describes the 'what' of a system, not the 'how'.
* This phase produces a large document, which contains a description of what the system will do without describing how it will be done. The resultant document is known as Software Requirement Specification (SRS) document.

(2) Design phase : The SRS document contains the exact requirements of the customer.

* The goal of this phase is to transform the requirement specification into a structure that is suitable for implementation in some programming language.
* The work is documented and is known as Software design Description (SDD) document.

(3) Implementation and unit testing phase :- During this phase, design is implemented.

* If the SDD is complete, the implementation and the coding phase proceeds smoothly.
* During the testing, small modules are tested

in isolation from the rest of the software product
* problems are solved in this phase and modules are tested after writing some overhead code.

(4) Integration and System Testing phase:
* This is a very impt phase and testing is a very expensive activity.
* Purpose of unit testing is to determine that each independent module is correctly implemented. This gives little chance to determine that the interface b/w modules is also correct and for this reason integration testing is performed.
* System Testing involves the testing of the entire system.

(5) Operational and maintenance phase: It is the task that every development group has to face when the software is delivered to the customer's site, installed and is operational.
* software maintenance includes error correction, enhancement of capabilities, deletion of obsolete capabilities and optimisation.

Disadvantages of Waterfall Model

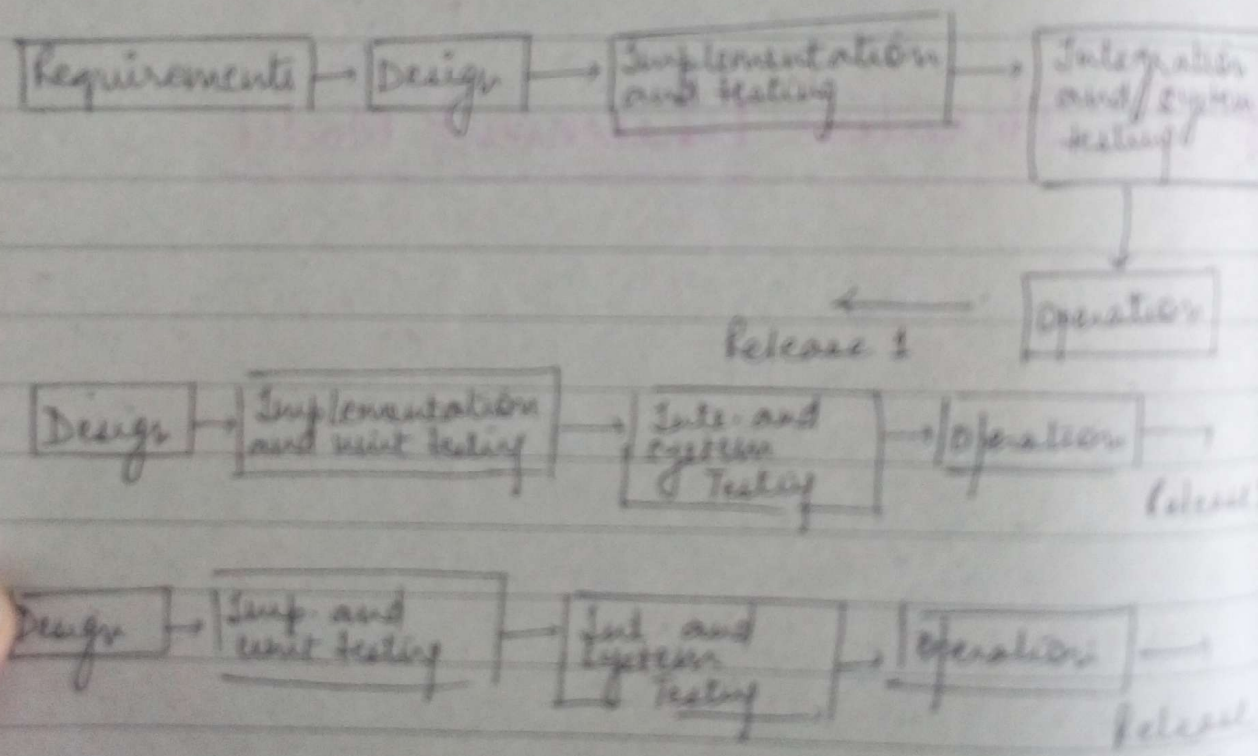
1. It is difficult to define all requirements at the beginning of a project
2. This model is not suitable for accomodating any change
3. It does not scale up well for large projects
4. End product can be viewed at the last phase only. Customer cannot see the desired software inbetween.

II Iterative Enhancement Model

* This model has the same phases as the waterfall model but with fewer restrictions
* During the first requirement analysis phase, customer and developers specify as many requirements as possible and prepare a SRS document
* Developers implement the specified requirements in one or more cycles of design, implementation and test based on the defined priorities
* This model does not deliver any operational quality product at each release, but only that satisfies a subset of the customer's requirements

* The complete product is divided into releases and the developer delivers the product release by release. A typical product will usually have many releases.

* With this model, the first release may be available within few weeks or months, whereas the customer generally waits month or years to receive a product using waterfall or prototyping model.



Page No.
Date

Prototype Model

* It is an alternative to waterfall model.
* In Prototype model we first develop a working prototype of the software instead of developing actual software. The working prototype is developed as per current available requirements.

* The developers use this prototype to refine the require-ment and prepare the final specification document. When the prototype is created, it is reviewed by the customer. This review gives feedback to the customer that helps to remove uncertainities in the require-ment of the software.

* The prototype may be a usable program, but is not suitable as the final software product because of poor performance, maintainability or overall quality.

* After the finalisation of software requirement and SRS document, the prototype is discarded and actual system is then developed using a waterfall approach. Thus, it is used as an input to waterfall model and produces maintainable and good quality software.

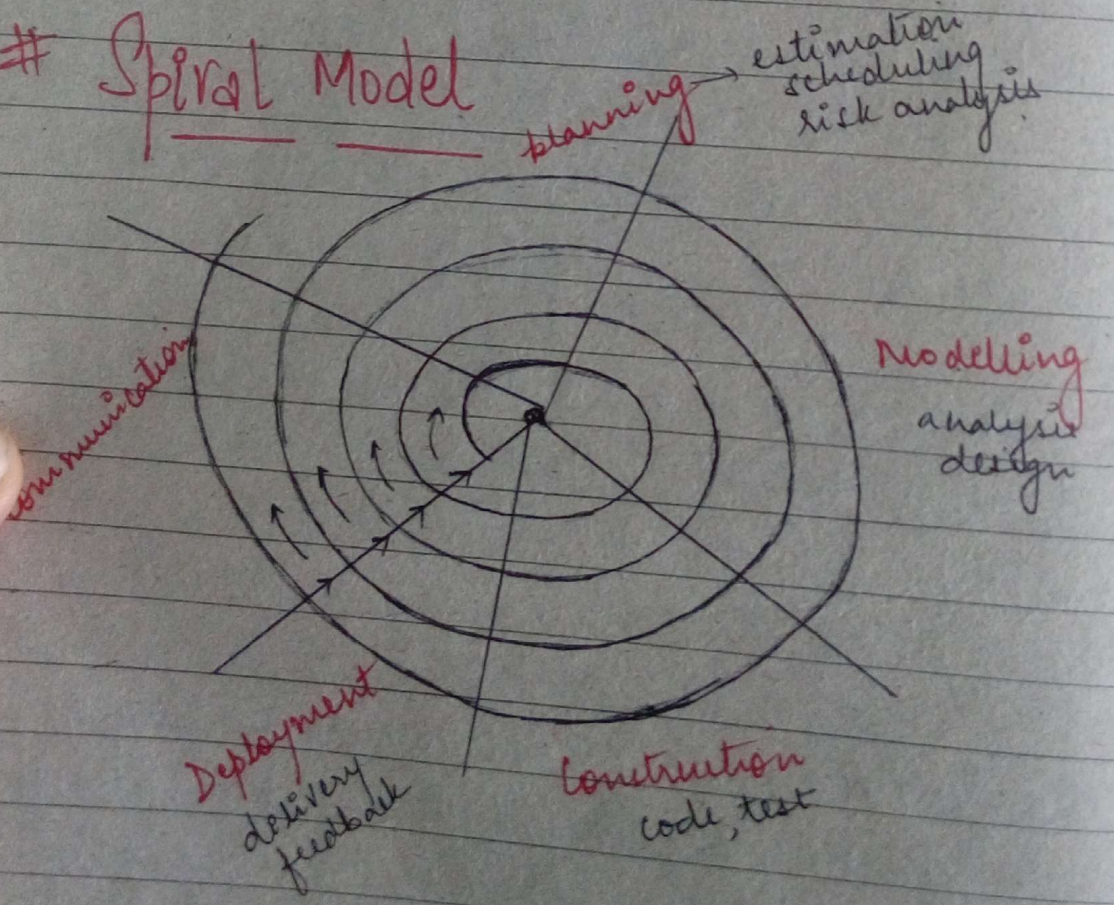
Page No.
Date

Advantage: Customer can view the prototype in between to check the requirements and to give his feedback.

Disadvantage: ① The development of a prototype might involve extra cost.

② This model requires extensive participation and involvement of the customer, which is not always possible.

Spiral Model



Page No.
Date

* Spiral model is also called Risk-driven Model
* The focus of Spiral model is ~~to~~ the identification of problems and the classification of these into different level of risks, the aim being to eliminate high risk problems before they threaten the software operation or cost.

* During the first phase, planning is performed, risks are analysed, prototypes are build and customer evaluate the prototype.

* During the 2nd phase, a more refined prototype is built, requirements are documented and validated and customers are involved in assessing the new prototype.

* By the 3rd phase begins, risks are known and more traditional development approach is taken.

* An important feature of spiral model is that, each phase is completed with the review by the people concerned with the project. The review consists of all the products developed upto that point and includes the plan for the next cycle.
* If the plan for the development fails, then the spiral is terminated otherwise, it terminates with the initiation of new or modified software.

Page No.
Date

Advantage :① * wide range of ~~features~~ options to accomodate the good features of other life cycle model.

② Risk analysis and validation steps eliminate errors in the early phases of development.

Rapid Application Development (RAD) Model

This model is an incremental process model and was developed by IBM in 1980's.

* The process is started with building a rapid prototype and is given to user for evaluation. The user feedback is obtained and prototype is refined. The process continues till the requirements are finalised.
* SRS and SDD are prepared with association of users.

* There are 4 phases in this model :-

① Requirement Planning Phase : Requirements are gathered.

② User description : Joint teams of developers and users are constituted to prepare, understand & review the requirements.

— Performance testing
— testable software
Characteristics

Page No.
Date

③ Construction phase : This phase combines the detailed design, coding and testing phase of waterfall model.
* Here, we release the product to the customer.

④ Cut over phase : This phase incooperates acceptance testing by the users, installation of the system and user training.

ADVANTAGE : ① Quick initial views about the product are possible due to delivery of rapid prototype.
② Involvement of the user may increase the acceptiblity of the product.

DISADVANTAGE : ① If user cannot be involved throughout the life cycle, this maynot be an appropriate model.
② Highly specialised & skilled developers are expected and such developers may not be available easily.

—x—

~~chapter 3~~

~~Software Requirements Analysis & Specification~~

~~Chapter 6~~: ~~function - Oriented Design~~

Software Testing

→ Software Testing is the process of testing the software product. Effective software Testing will contribute to the delivery of higher quality software products, more satisfied user, lower maintenence cost, more accurate and reliable product.

→ Software Testing is neccessary and important activity of software development process.
→ It is very expensive process and consumes 1/3 of the cost of development project.

Page No.
Date

Testing is the process of executing a program with the intent of finding errors.

* When we want to test a program, we want to add some value to the program. Adding value means raising the quality or reliability of the program.
* Raising the reliability of the program means finding and removing errors.

Hence we should not test a program to show that it works; rather we should start with the assumption that the program contain errors & then test the program to find as many errors as possible.

ERROR	DEFECT
It is the one which is generated because of wrong login, loop or due to syntax. Error means to change the functionality of the program.	It is the difference b/w expected & actual result in the context of testing. It refers to several troubles with the software product with its external behaviour of with its internal features
* Caused before the	* Caused after the release

release of software | of software.

Bug: It is an error, flaw, mistake or fault in a computer program that prevents it from working as intended or produces an incorrect result.

* It arises from the mistakes and errors made by people in either program's source code or its design.

Validation	Verification
* It is the process of evaluating a system or component during or at the end of development process to determine whether it satisfies the specified requirements.	* It is the process of evaluating a system or component to determine whether the products of the given development phase satisfy the conditions imposed at the start of the phase.

* It describes whether the software is accepted by the user or not.	* It describes whether the outputs are according to inputs or not.
* It always involves executing the code.	* It doesnot involve executing the code.
* It is program based execution of program.	* It is human based checking of documents and files
* Validation activity is carried out just after the verification	* Verification is carried out before the validation
* It involves all dynamic testing techniques.	* It involves all static testing techniques
* eg: all types of testing like smoke, regression etc.	eg: includes reviews, inspection etc.

Alpha testing	Beta testing
(1) It is always performed by the developers at software development site	(1) It is always performed by the customers at their own site
(2) Alpha testing is not open to the market & public	(2) Beta testing is always open to the market & public
(3) It is usually performed in Virtual environment	(3) It is usually conducted in real environment
(4) It is always performed within the organisation	(4) performed outside the organisation
(5) It comes under the category of both black box & white box testing	(5) It is only a kind of black box testing
(6) developers are present	(6) Developers are not present.

Boundary Value Analysis:

* It is the most common and impt. technique for test case design.
* It is based on testing at the boundaries b/w partition
* BVA is where test cases are generated using the extremes of the input domain. eg: max, min, just inside/outside boundary value.
* BVA is a test design & technique that is used to find the errors at boundaries of input domain rather than in the centre of the input.

Disadvantage: ① cannot be used for boolean and logical variables

Advantage: results in small set of test cases that can be easily covered by any software test analyst.

BVA is based on 'single fault assumption". This says that failures are rarely the result of simultaneous occurence of 2 or more faults.

* BVA test cases are obtained by holding the values of all but one variable at their nominal values and letting that variable assume its extreme values.
* Test cases should be created to generate inputs or outputs that will fall on and to either side of each boundary, which results in 2 cases per boundary.

Eg: Consider a program with 2 input variables x & y. These input variables have specified boundaries as:

$$a \leq x \leq b$$
$$c \leq y \leq d$$

Hence, both the inputs x and y are bounded by 2 intervals $[a,b]$ and $[c,d]$ for input x, we may design test cases with values a and b, just above a and below b.

Similarly for y, just above c and below d!

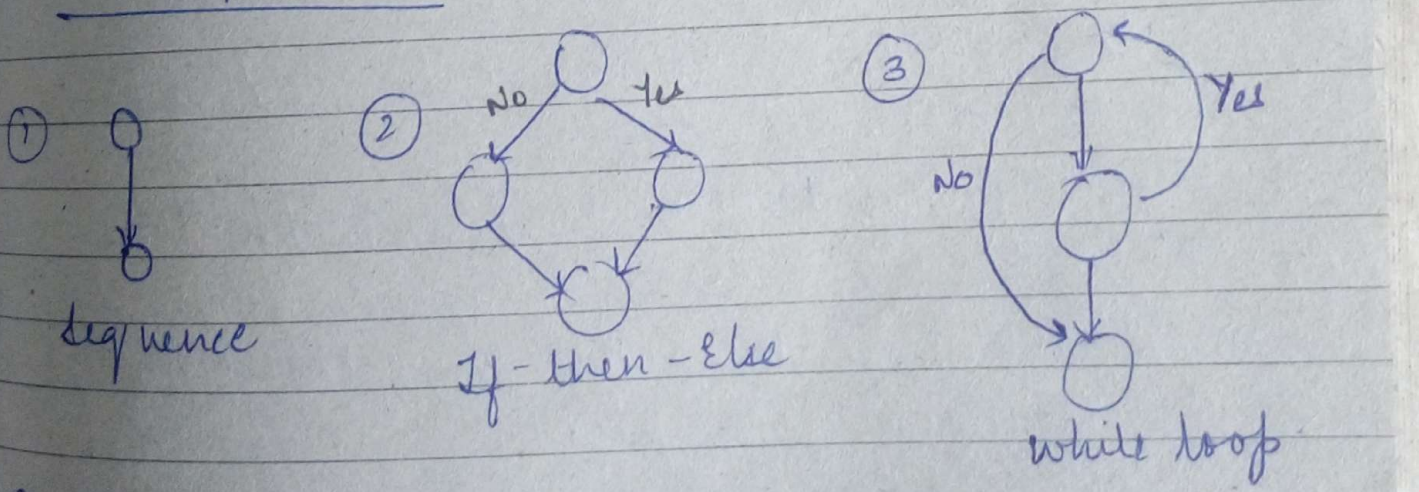
Page No.
Date

Structural Testing :

- ~~type of functional Testing~~
- ~~adhoc Testing~~

→ complementary to functional Testing / whitebox.
→ permits to examine internal structure of program.
→ knowledge of internal structure is used to find test cases.

Flow Graph :



sequence

If - then - Else

while loop

repeat until loop

Switch Statement

Cyclomatic Complexity:

- also known as structural complexity because it gives internal view of the code.
→ used to find the no. of independent paths through a program.
- Gives the upper bound for the no. of tests.
- Statements have been executed atleast once.

I $V(G) = \cancel{e+n}\ e - n + 2P$ → connected components

e → edges, n → vertices

→ graph has a unique entry and exit nodes.

→ graph is known as flow graph.

→ 2 alternate methods for calculating Cyclomatic complexity.

(1) Cyclomatic complexity $V(G)$ is equal to the no. of predicate nodes + 1

$$V(G) = \pi + 1$$

π = predicate node.

→ Every predicate node should have 2 outgoing edges, ie one for 'true', other for 'false' condition.

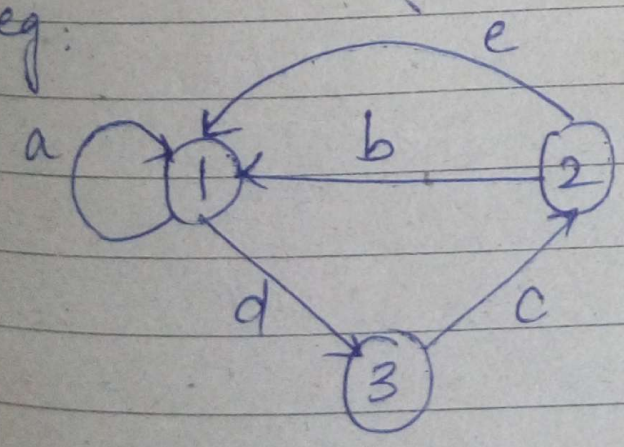
(2) Cyclomatic complexity is equal to the no. of regions of the flow graph.

Graph matrices :

→ It is a square matrix with one row and one column for every node in the graph.

→ The size of the matrix is equal to the no. of nodes in the graph.

eg:



	1	2	3
1	a		d
2	b+e		
3		c	

SQA

(1) Is it possible to asses the quality of s/w if the customer keeps changing the functional requirements?

(2) What do you understand by software Metrices. Discuss the various product metrics and project matrix for s/w development.

(3) 4 measures of s/w Quality.

(4) Various matrices of s/w Quality

(5) 3 roles of SQA group

(6) Umbrella activities.

(7) What is SQA? What are the various activities carried out by SQA team.

(8) Explain Defect Amplification & Removal Model

Page No.
Date

Ans 7. SQA is a process that ensures that developed s/w meets and complies with defined and standardised quality specifications.

SQA is an ongoing process within the SDLC that routinely checks that developed software ~~to ensure~~ it meets desired quality measures.

Rather than checking for quality after completion, SQA process test for quality in each phase of development until s/w is complete.

Activities involved:

1. formulating a quality management plan
2. Applying s/w engg techniques
3. controlling change
4. keeping records and reporting

Ans 8 (5) Main activity is to develop

Page No.
Date

Ans 7. SQA is a process that ensures that developed s/w meets and compiles with defined and standarised quality specifications.

SQA is an ongoing process within the SDLC that routinely checks that developed software ~~to ensure~~ it meets desired quality measures.

Rather than checking for quality after completion, SQA process test for quality in each phase of development until s/w is complete.

Activities involved:

1. formulating a quality management plan
2. Applying s/w engg techniques
3. controlling change
4. keeping records and reporting

Ans 5) Main activity is to develop

DRE

① DRE ? How does it assist the developer to improve software quality.

② How does it is used to access the team's availability to find the errors as they are passed to the next framework activity.

Ans: DRE gives the measure of development team ability to remove defects prior to release. It is calculated as the ratio of defects resolved to total no. of defects found.

$$DRE = \frac{E}{E+D}$$

E = No. of errors found before delivery of the s/w

D = No. of defects found after delivery.

Ideal value of DRE = 1. (No defects found)

DRE is a measure of filtering ability of QA Activities as they are applied throughout the framework.

DRE encourages a s/w team to institute techonology for finding as many errors as possible as before delivery.

(2) Requirement analysis produces a requirement model that can be reviewed to find and correct errors. Those errors are not found during the reviews of the requirement model are passed onto design.

$$DRE_i = \frac{E_i}{E_i + E_{i+1}}$$

E_i = No. of Errors found duriy software engg action i and E_{i+1} = No. of errors found duriy s/w engg action E_{i+1}.
The quality objective of s/w team is to achieve DRE_i that approaches 1. Ie, errors should be filtered out before they are passed on to next activity or action.