2 Software Requirement Engineering

IMP

* Core principles of software engineering

1. Principles : The reason it all exists :-
A software system exists for one important reason to provide value to its users. All decisions of a software system should be made with this in mind

2. Principle :- keep it simple stupid (KISS) There are many terms and factors to consider in any software design effort, All software design should be as simple possible, but no simpler. this concept facilitates having a more easily understood and easily maintained software system.

3. Maintain the vision :- A clear vision is essential and important to the success of a software project

4. what you produce, other will consume :- always specify, design and ~~impor~~ implement knowing someone else will have to understand what you are doing.

5) DRY (Dont repeat yourself):-
Avoiding duplication of code and data in the soft. This makes the softwar more important maintainable and less error-prone

6) Abstraction :- Hiding the implementation details of a ~~de~~ module or component and exposing only the necessary ~~in~~ information This makes the softwar more flexible and easier to change.

7) Modulority :- Breaking down the softwar into smaller, independent, and reusable components or moduks, This makes the softwar easier to understand, test, and maintain.

Communication Practices principles.

1. Principle #1 :- Listen carefully :- To understand customer's requirments carfully and perfectly. Listening principle ensures the proper data collection from the speaker. You should ask ~~those st~~ questions for doubts mind if you have get them clarifies.

2. Principle #2 :- Preparation befor communication :- It is very important to prepare the agenda for the meeting or discussion. The Agenda contains the point to be discussed in the meeting.

3. Principle #3 : face-to-face communication :- it is better even if some documents related to that particular discussion is given.

4. Take notes and document decision :- write down all important points and issues raised in the meeting discussion.

5. Draw a picture if something is unclear :- Verbal communication goes only in this principle. A drawing can often provide clarity when words fails to do the job.

* Planning Practices * Principles

1) Principle # 1 :- To understand the scope of the project :- scope of project provides the software project team with a destination.

2) To involve the customer in the planning activities : The customer defines software project priorities and establishes project constraints.

3) Be realistic :- Every member of software project team do not work 100 percent of every day. Basically, noise always enters into any human communication.

4) To consider risk as you define the plan : If the software project team has defined risks that have high probability and impact, contingency planning is necessary.

1. Softwar Deployment :- Principles

1. Principle #1 :- Manage customers expectations or requirements :- In most cases customer wants more than he has stated earlier as his requirements or expectations. In many cases customer is disappointed, even after getting all his requirements satisfied. Hence at the time of softwar delivery, developer must have skills to manage the customer's expectations and requirements.

2. Record keeping mechanism must be established for customer support :-
:- 'customer support' is essential ~~must~~ and important factor in deployment phase. The support should be well planned and with proper record-keeping mechanism.

3. Provide essential instructions, documentations and manual :- Actual softwar project delivery includes all documentations, helps files and guidence for handling the softwar by user.

4. Assemble and test complete package :- The customer side must get all supporting and essential help from developer side for this reason CD with complete assembled and tested delivery package should be delivered.

* concept of SRS
* SRS (
* A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development.
* The SRS fully describes what the software will do and how it will be expected to perform.

* Requirements documents are essential when an outside contractor is developing the software system. The software requirements specification (SRS) is a technical specification of requirements for the software product.

* Depending on the size and complexity of the software product, the SRS may consists of a few pages and its based on the system definition.

* features of SRS

1. It forms the basis for software development
2. SRS ~~provides~~ helps to clients and ~~used~~ understand their own needs and requirments.
3. It establishes the basis for agreement between the client and the supplier.

IMP

Need / importance of SRS

- There are three major parties interested in a system : the client, the users and the developer. somehow the requirements for the system that will satisfy the needs of the clients and the concerns of the users have to be communicated to the developer. The problem is that the client usually does not understand software or the software development process and the developer often does not understand the clients problem and application area. This causes a communication gap between the parties involved in the development project. A basic purpose of software requirements specification (SRS) is to bridge this communication gap. SRS is the medium through which the client and user needs are accurately specified; indeed SRS forms the basis of software development.

- Another important purpose of developing an SRS is helping the clients understand their own needs or requirements.

- An SRS is important because it establishes the basis for agreement between the clients and the ~~sd~~ suppliers on what the software product will do.
- SRS provides a reference for validation of the final product.

* A high-quality SRS is a prerequisite to high-quality software.

* A high-quality SRS reduces the development cost.

imp
* Characteristics of SRS:

- software requirements specification should be accurate, complete, efficient, and of high quality, so that it does not affect the entire project plan. An SRS is said to be high quality when the developer and user easily understand the prepared document.

1. Correct: SRS is correct when all user requirements are stated in the requirements document. The requirement should be according to the desired system. This implies that each requirement is examined to ensure that it (SRS) represents user requirements.

2. complete: SRS is complete when the requirements clearly defines what the software is required to do. This includes all the requirements related to performance, design and functionality

3. ~~Modifiable: the requirements are not equally important~~

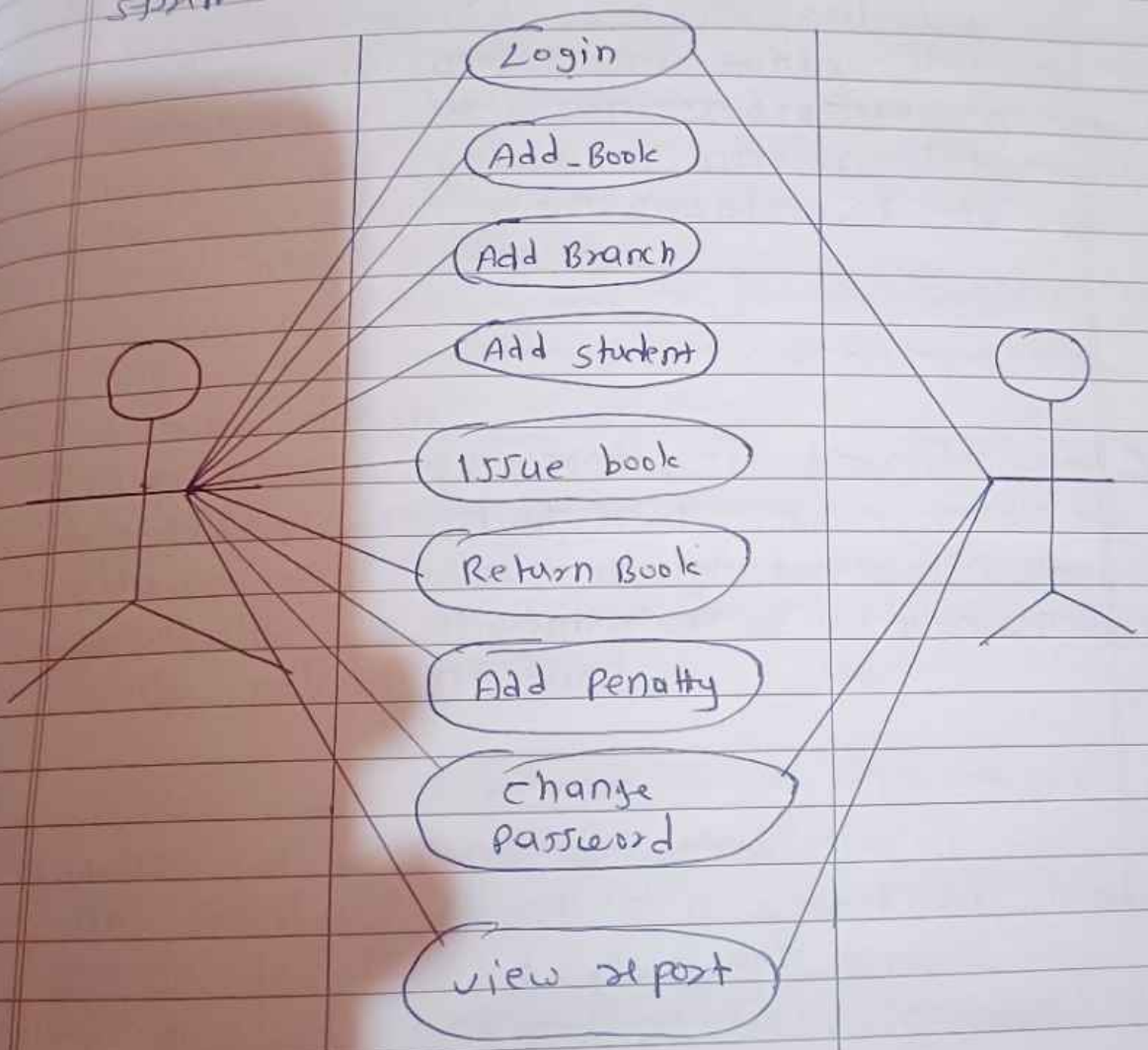
2) Modifiable :- The requirements of the user can change, hence requirments document should be created in such a manner that those changes can be modified easily.

4) Stability :- All requirments are not equally important, hence each requirments is identified to make differences among other requirments. For this, it is essential to clearly identify each requirment. stability implies the probability of changes in the requirment in future.

5) Consistent :- SRS is consistent when the subsets of individual requirments defined do not conflit with each other.

6) Verifiable :- SRS is ~~trac~~ verifiable when the specified requirments can be verified with a cost effective process to check whether the final software meets those requirments.

* Use case diagram for library management system.



14 MARK

3] software modeling and design

1] Requirment Analysis :-
It is concerned with the collection of information about the system. The information helps in understanding various aspects of operations and in determining the functional requirements of the system and the software.

x] software design
→ It produces a specification of how the input-output relationships specified in the SRS document are obtained. The software design document (design specification) is the blueprint of the proposed solution to the problem at hand.

IMP
*] what is analysis modeling.
∴ An analysis model is a structural model that is crated fom a physical model It is used for analyzing structural behavior and load bearing, and for design.

*] what is analysis model.
→ Analysis model is a technical representation of the system. It acts as a link between system description and design model.-

IMP

* Cardinality and modality

* cardinality :- It specifies the number of occurrences (instances) of one data object or entity that relates to the number of occurrence of another data object or entity.

- It also specifies the number of entities that are included in a relationship.

* Modality :- It describes the possibility whether a relationship between two or more entities or data objects is required.
- The modality of a relationship is 0 if the relationship is optical.

For example :- a husband can have only one wife; while a parent can have many children.

* ~~Advantage of Data flow Diagram~~

very IMP

* Data
- Th
re
inf
pro
vie
- A
- A
for
lik
da
ro
- DF
in
da
- Ad
- A
- A
- Di

Very IMP

* Data Flow Diagrams (DFDs)

- The flow Diagram (DFD) is a graphical representation of the flow of data through an information system. It enables us to represent the processes in the information system from the viewpoint of data.

- A data flow diagram also known as ~~a bubble~~ a bubble chart or ~~flow~~ work flow diagram

- A DFD maps out the flow of ~~info~~ information for any process or system. It uses defined symbols like rectangle, circles and arrows, to show data inputs, outputs, storage points and the routes between each destination.

- DFD focus on the process that transforms incoming data flow (input) into outgoing data flows (output).

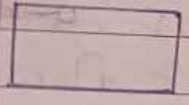
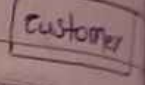
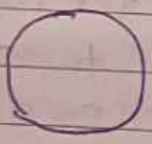
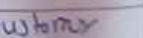
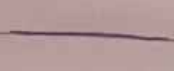
- Advantages of DFD
- A simple graphical technique which is easy to understand
- A DFD explains the logic behind the data flow within the system.

- Dissadvantages of DFD

- The DFD takes a long time to create
- Physical considerations are left out in DFD

* ~~Data Flow Diagrams (DFDs)~~

- Symbols used in DFDs with example:

Name	symbol	Description	Example
Entity		used to represent people and organizaitons outside the system. they either input information to the system.	
Process		A process accepts inputs data and produces data that it passes on to another part of the DFD	
Data flow		These represents the flow of data to or from a process	
Data store		This is place where data is stored either is temprary or Permanenty.	Products

. Draw DFD for Bank system:

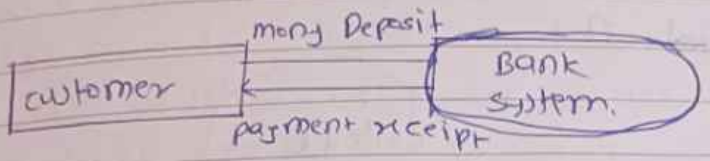


Fig: 1 : Level - 0 DFD of Bank system.

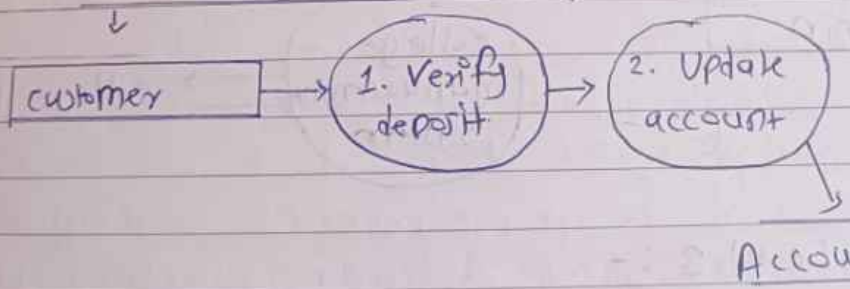
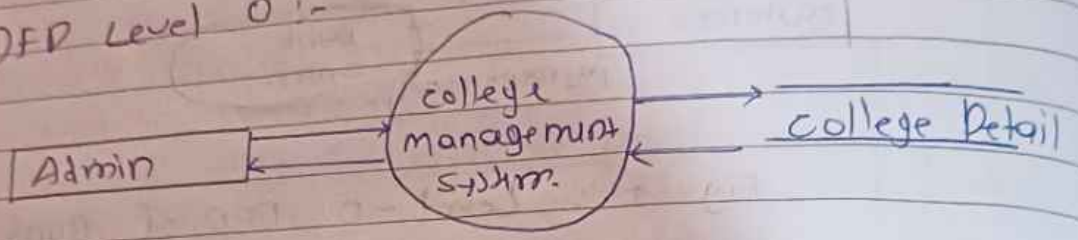


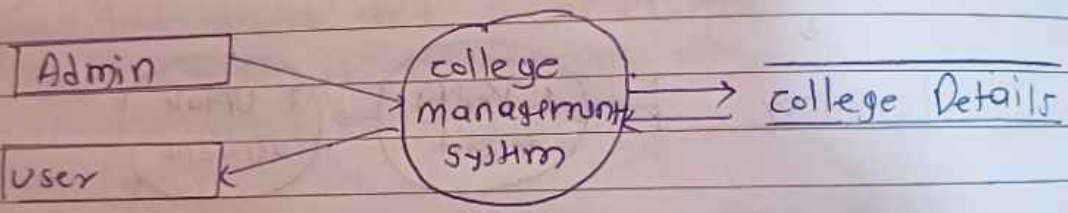
fig: 2 : Level - 1 DFD of Bank system.

* Draw DFD for college management system.

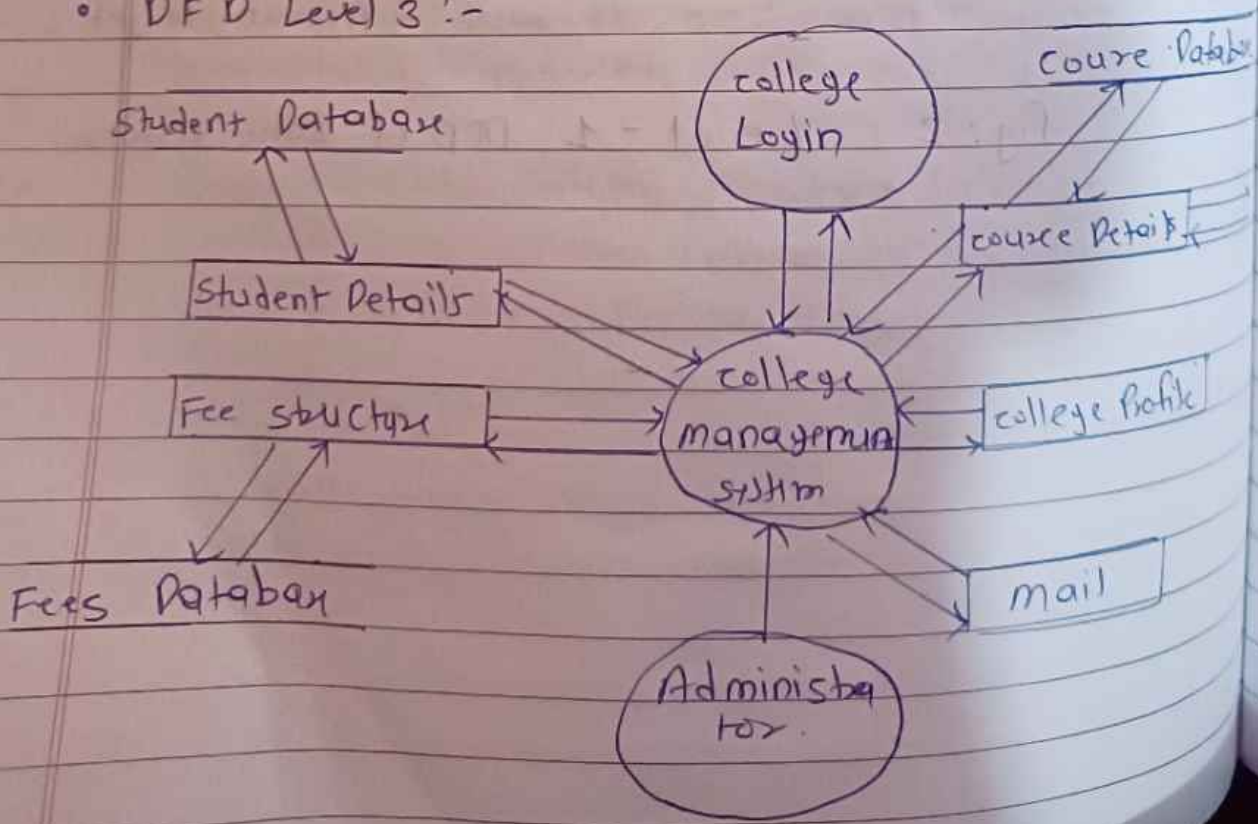
- DFD Level 0 :-



- DFD Level 1 :-



- DFD Level 3 :-



IMP

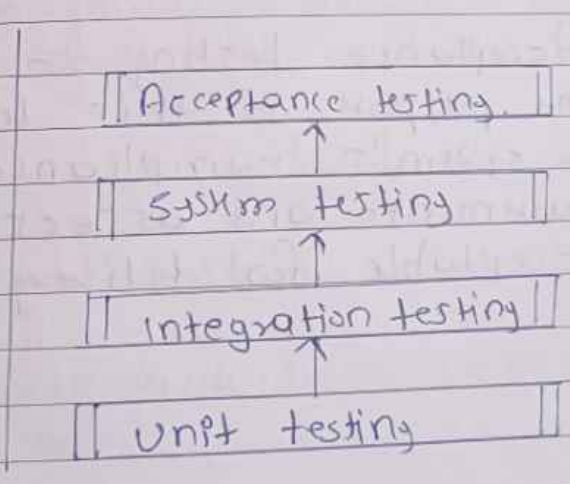
* Testing :-

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

* Levels of testing

→ . A level of software testing is a process where every unit or component of a software / system is tested.

. software testing has various levels. At each level there are specific testing goals. There are four levels of software testing as shown



→ college Database

1] Unit testing :- A level of the software testing process where individual units of a software are tested. the purpose is to validate that each unit of the software performs as designed. The idea of this is to confirm every parts or unit of the product after the test.

2. Integration Testing :- A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

3. System testing :- The purpose of this test is to evaluate the system's compliance with the specified requirements.

4) Acceptance testing :-

→ The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Testing characteristics

1) Testing is a process of executing a program with the intent of finding an error

2) A good test case is one that has a high probability of finding an as yet undiscovered error

3) A successful test is one that uncovers an as yet undiscovered error

4) Testing cannot show the absence of defects it can only show that software defects are present.

* attributes of "good test"

1) A good test has a high probability of finding an error

2) A good test should be neither too simple nor too complex

3) A good test should be "best of breed"

4) A good test is not redundant.

Very important

Difference between verification and validation

Verification	Validation
verification is doing things right	Validation is doing right things.
It is process based	It is product based
verification takes place first and includes the checking for documentation, code.	validation occurs after verification and mainly involves the checking of the overall product
Done by developers	Done by testers.
It ensures that the software system meets all the functionality	It ensures that the functionalities meet the intended behavior

Testing Methods :-

Once the software is developed it should be tested in a proper manner before the system is delivered to the user. For this techniques that proper systematic guidance for designing tests are used.

1) Black box testing
2) white box testing

1) Black box testing.

:- Black Box testing checks the functional requirments and examines the input and output data of these requirments.

- The purpose of Black Box testing to test the functionality of the site.
- when Black Box testing is performed only the sets of 'legal' input and corresponding outputs should be known to the tester and not the internal logic of the program to produce that output. Hence to determine the functionality the outputs produced for the given sets of input are observed.

- Black Box testing also known as behavioral testing. is a software testing method in which the internal structure / design / implementation of the item being tested is not known to the tester
- In Black Box testing the tester is concentrating on what the software does not how it does it.

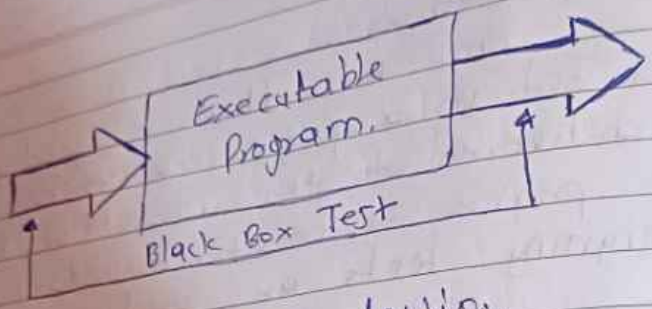


fig:- Black Box testing.

Advantages of ~~Box~~ Black Box testing

- In Black Box testing the code access is not required.
- This testing is well suited and efficient for large code segments.

Disadvantages of Black Box testing.

- In this testing the test cases are difficult to design.
- Inefficient testing due to the fact that the tester only has limited knowledge about an application.

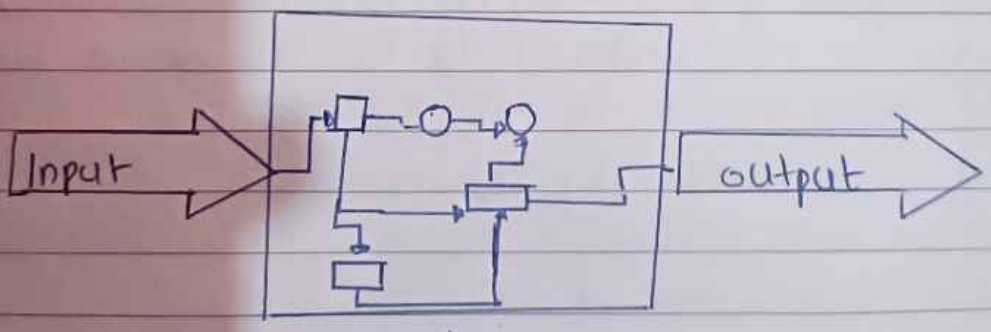
White Box testing.

- White Box testing is also called structural testing and Glass Box testing. White Box is a testing technique that takes into account the internal mechanism of a system or component.

- In this testing, test the artifact from internal point of view internal structure of the application.

- White Box test cases cannot be started in the initial phase of the project because it needs more architecture clarity which is not available at the start of the project. These test cases are written after black box test cases are written.

- White Box needs more structural understanding and knowledge of the software structure and implementation.



White Box testing.

- Advantages of white box testing
- It helps in optimizing the code.
- white box testing are easy to automate.

- Dissadvantages of white box testing.
- It is difficult to maintain white box testing
- sometimes It is impossible to look into every nook and corner. to find out hidden errors.

	Black Box testing	White Box testing.
1	The internal working of an application need not be known	Tester full knowledge of the internal workings of the application.
2.	Also known as closed Box testing data drive testing or functional testing	Also known as clear box testing, structural testing or code-based testing.
3.	Performed by end-users and also by testers and developers	Normally done by testers and developers.
4.	It is exhaustive and the least time consuming	The most exhaustive and time-consuming type of testing.
5	Not suited for algorithm testing	suited for algorithm testing.
6	This can only be done by trial-and-error method	Data domains and internal boundaries can be better tested.

- Testing techniques

- Unit testing.
- Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level.
- It is written by developers as they work on code to ensure that the specific function level
- It is used to ensure that the building blocks of the software work independently from each other. used to reduce software development risks, time and costs.

- It is performed by the software developer or engineer during the construction phase of the software development lifecycle.

It is intended to increase the quality of the resulting software as well as the efficiency of the overall development and QA (quality Assurance) process.

Types of Acceptance testing
1] Alpha testing
2] Beta testing

1) Alpha testing :- • Alpha testing is conducted by customer's site. It is performed by potential users like developer, end users or organization users before it is released to external customers and report the defects found while Alpha testing.
• This software product testing is not final version of software application, after fixing all reported bug (after bug triage) the new version of software application will release.
• Sometimes the alpha testing is carried out by client or an outsider with the attendance of developer and tester.

* Advantages of Alpha testing:
• It identifies all the errors present in the software.
• It provides better view about the reliability of the software at an early stage.

* Disadvantages of alpha testing.
• In depth functionality cannot be tested as software is still under development stage.
• Sometimes developers and testers are dissatisfied with the results of alpha testing.

2. Beta testing
- Beta testing is acceptance testing conducted at the customer end. Beta testing is system testing performed by selected group of customers.
- Beta testing is to be carried out without any help of developers at the end user's site by the end users and so it is performed under uncontrolled environment.
- Beta testing is also known as field testing this is used to get feedback from the market
- This testing is conducted by limited users and all issues found during this testing are reported on continuous basis which helps to improve the system.
- this is last stage of the testing where product is sent outside the company or for trial offer to download.

* Advantages of beta testing.
- reduces product failure risk via customer validation
- improves product quality via customer feedback

* Disadvantages of beta testing.
- Test management this testing is difficult to manage.

Alpha testing	Beta testing.
• It is always performed by the developers at the software development site	• It is always performed by the customers at their own site.
• Alpha testing is conducted by a team of highly skilled testers at development site	Beta testing is always conducted by customers or end users at their own site.
• Alpha testing is not open to the market and public	• Beta testing is always open to the market and public
• It is conducted for the software application and project.	It is usually conducted for software product.
• It is always performed in virtual environment	• It is performed in real time environment.
• It is always performed within the organization	It is ~~also~~ always performed outside the organization
It comes under the category of both white box testing and Back Box testing	It is only kind of Black Box testing
Long execution cycle may be required for alpha testing	only few weeks of execution are required for Beta testing

4: Software project Estimation:

* Introduction:
- Estimation is the process of finding an estimate or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable.
- Estimation determines how money, effort, resources and time it will take to build a specific system or product.
- Software cost estimation is a critical part in the development of a software project, and over the years has become an Emerging research area.

IMP

① The management spectrum (4P's)
:- Software engineering is concerned with both technical as well as managerial aspects of software development. There are four important core aspects of software development these four (4P's) aspects are
① People
② Product
③ Process
④ Project

① The People :- · Software development requires creativity and knowledge work. People of a project includes from manager to developer, from customer to end user. But mainly people of a project highlight the developer

- Organizations that achieve high levels of maturity in the people management area have a higher likelihood of implementing effective software engineering practices.

② The product :- Software as a product has to perform certain specific functions required by users (or customers) Determination of correct functional requirements and features of software to be produced is a very critical activity of software development. The product in the context of software is the scope of the software that is proposed to solve the requirements of the user

③ The process :- A software process provides the framework from which a comprehensive plan for software development can be established.

- In general, a process is defined as a series of steps involving activities and resources which produces the desired output. Software development process is defined as a collection of procedures to develop the software product according to certain goals or standards.
- The process specifies the policies procedures tools and techniques to be used for software development. A number of models are available such as CMM (capability maturity model) is a widely used standard model for software development process.

6) The project :- The project includes all ons everything of the total development process and to avoid project failure the manager has to take some steps, has to be concerned about some common warnings etc.

- A project a specification essential for developme or maintaining a specific product, Software project is developed when the software processes are executed for certain, specific requirments of the user.

* Reasons why some projects fails

i) Unrealistic project schedules
ii) Unclear project goals
iii) Inadequate communication skills
iv) inadequate administrative skills

* Lines of code (LOC):-

- LOC is the simplest among all metrics available to estimate project size. This metric is very popular being the simplest to use
- LOC is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program source code
- Lines of code (LOC) is one of the most widely used methods for size estimation. LOC is defined as, "the number of delivered lines of code, excluding comments and blank lines."

- LOC has also been used to predict program complexity, development effort, programmer performance, and so on.

* Function Point (FP) :-

- FP is developed in 1979 by Allan J. Albrecht of IBM. function point is a metric that measures the software size as a magnitude of the functionality delivered by the system to its users.

- Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value the most widely used function-oriented metric is the function point (FP).