Modern Answer Sheet Class II Software Engineering

Q.1

1. List the objective of black box testing.

1. Black-box tests are used to demonstrate that software functions are operational
2. Input is properly accepted and output is correctly produced,
3. The integrity of external information (e.g., a database) is maintained.
4. A black-box test examines some fundamental aspect of a system with little regard for the internal logical structure of the software.

2. Explain qualities of software considering.

i) Quality of design

ii) Quality of conformance

i) Quality of design:

Quality of design refers to the characteristics that designers specify for a product. The grade of materials, tolerances, and performance specifications all contribute to the quality of design. As higher-grade materials are used, tighter tolerances and greater levels of performance are specified, the design quality of a product increases, if the product is manufactured according to specifications. In software development, quality of design encompasses the degree to which the design meets the functions and features specified in the requirements model.

ii) Quality of conformance:

Quality of conformance focuses on the degree to which the implementation follows the design and the resulting system meets its requirements and performance goals.

3. What do you mean by good test?

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

- A good test case is one that has a high probability of finding an as yet undiscovered error. A good test has a high probability of finding an error.
- A good test is not redundant.
- A good test should be "best of breed".
- A good test should be neither too simple nor too complex.
- A good testing strategy also assesses other quality characteristics such as portability, maintainability, and usability
- Each test should be executed separately; combining a series of tests could cause side effects and mask certain errors.

4. What is alpha-beta testing?

Alpha Testing: -The *alpha test* is conducted at the developer's site by a customer. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problems. Alpha tests are conducted in a controlled environment.

Beta Testing: - The *beta test* is conducted at one or more customer sites by the end-user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a "live" application of the software in an environment that cannot be controlled by the developer. The customer records all problems (real or imagined) that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta tests, software engineers make modifications and then prepare for release of the software product to the entire customer base.

Q.2

1.Explain the steps of bottom up integration.

Bottom-Up integration:

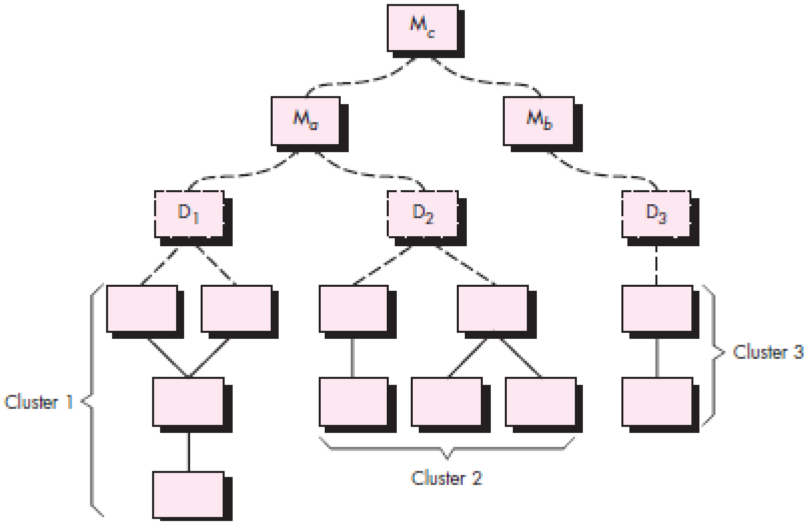
This begins with the construction a test of small modules. The components are integrated from the bottom-up, the functionally provided by components subordinate to a given level is always available and the need for stubs is eliminated. A bottom-up integration strategy may be implemented with the following steps:

1. Low-level components are combined into clusters that perform a specific software sub-function.

2. A driver is written to coordinate test case input and output.

3. The cluster is tested

4. Drivers are removed and clusters are combined moving upward in the program structure.



2 Explain the factors that Delay Project Schedule.

Although there are many reasons why software is delivered late, most can be traced to one or more of the following root causes:

1. An unrealistic deadline established by someone outside the software development group and forced on managers and practitioners within the group.

2. Changing customer requirements that are not reflected in schedule changes.

3. An honest underestimate of the amount of effort and/or the number of resources that will be required to do the job.

4. Predictable and/or unpredictable risks that were not considered when the project commenced.

5. Technical difficulties that could not have been foreseen in advance.

6. Human difficulties that could not have been foreseen in advance.

7. Miscommunication among project staff that results in delays.

8. A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem.

3. Enlist and explain different types of Software Risks.

1. Generic risks are a potential threat to every software project.

2. Product-specific risks can be identified only by those with a clear understanding of the technology, the people, and the environment.

3. Product size— risks associated with the overall size of the software to be built or modified that is specific to the software that is to be built.

4. Business impact— risks associated with constraints imposed by management or the marketplace.

5. Project risks threaten the project plan. That is, if project risks become real, it is likely that the project schedule will slip and that costs will increase.

6. Technical risks threaten the quality and timeliness of the software to be produced.

7. Business risks threaten the viability of the software to be built and often jeopardize the project or the product.

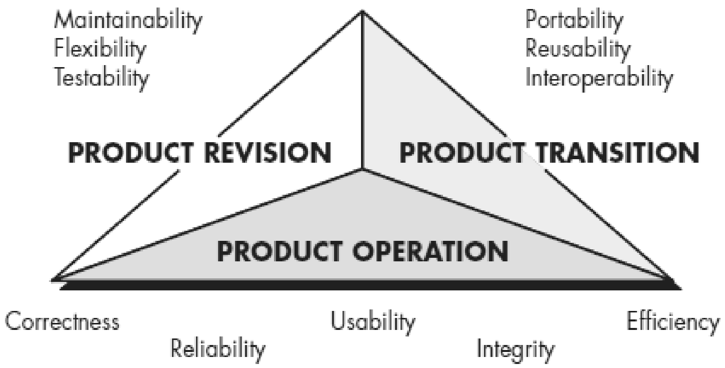
8. Known risks are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources.

9. Predictable risks are extrapolated from past project experience of user.

10. Unpredictable risks are one that they can and do occur, but they are extremely difficult to identify in advance.

Q.3

1.Explain Mccalls quality factor.



• Correctness. The extent to which a program satisfies its specification and fulfills the customer’s mission objectives.

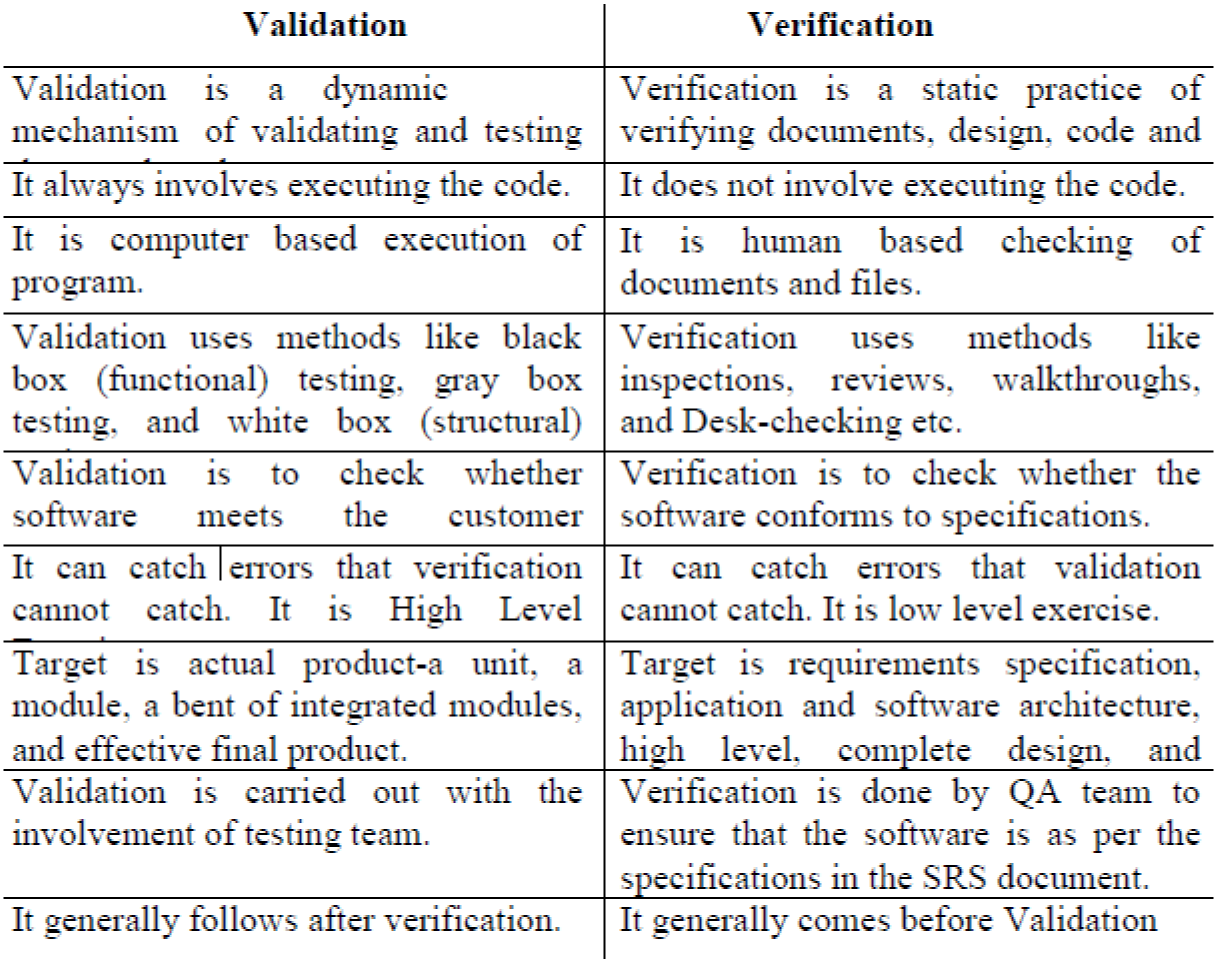
• Reliability. The extent to which a program can be expected to perform its intended function with required precision.

• Efficiency. The amount of computing resources and code required by a program to perform its function.

• Integrity. Extent to which access to software or data by unauthorized persons can be controlled.

- Usability. Effort required to learn, operate, prepare input for, and interpret output of a program
- Maintainability. Effort required to locate and fix an error in a program.
- Flexibility. Effort required to modify an operational program.
- Testability. Effort required to test a program to ensure that it performs its intended function.
- Portability. Effort required to transfer the program from one hardware and/or software system environment to another.
- Reusability. Extent to which a program [or parts of a program] can be reused in other applications— related to the packaging and scope of the functions that the program performs.
- Interoperability. Effort required to couple one system to another

2.Differentiate between validation and verification.



Validation	Verification
Validation is a dynamic mechanism of validating and testing	Verification is a static practice of verifying documents, design, code and
It always involves executing the code.	It does not involve executing the code.
It is computer based execution of program.	It is human based checking of documents and files.
Validation uses methods like black box (functional) testing, gray box testing, and white box (structural)	Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.
Validation is to check whether software meets the customer	Verification is to check whether the software conforms to specifications.
It can catch errors that verification cannot catch. It is High Level	It can catch errors that validation cannot catch. It is low level exercise.
Target is actual product-a unit, a module, a bent of integrated modules, and effective final product.	Target is requirements specification, application and software architecture, high level, complete design, and
Validation is carried out with the involvement of testing team.	Verification is done by QA team to ensure that the software is as per the specifications in the SRS document.
It generally follows after verification.	It generally comes before Validation