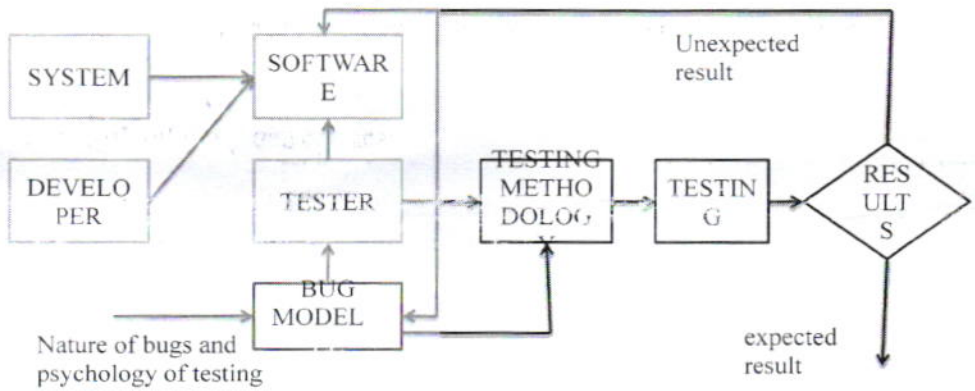
D 13

Scoring Indicators

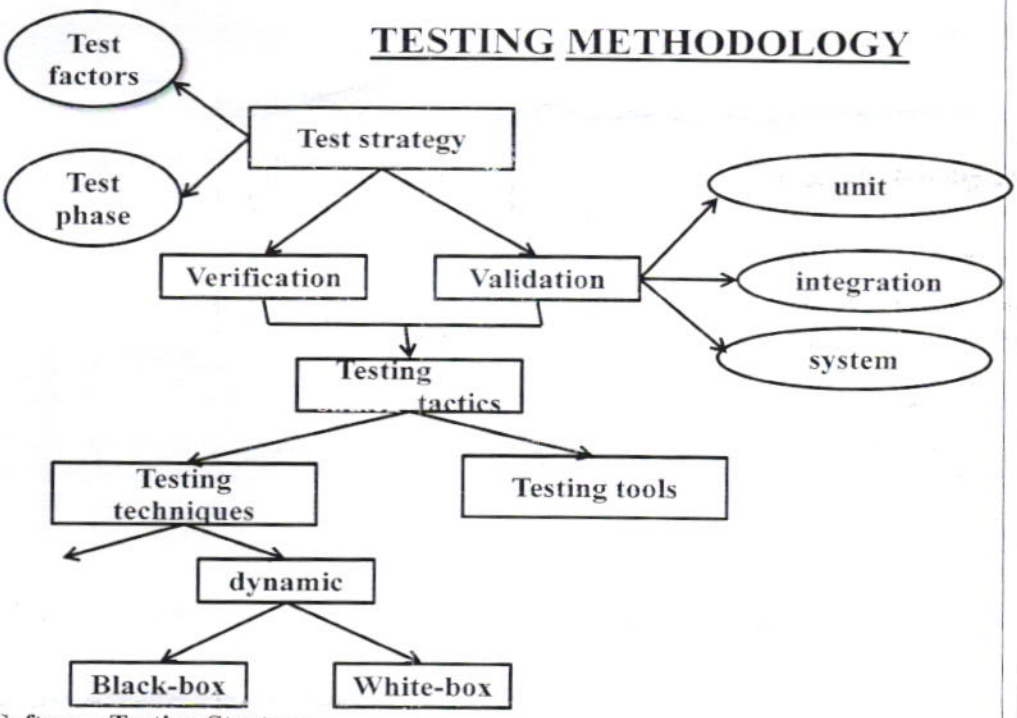
(15) 6136
Code TED(~~10~~)~~5069~~
Version A

Qn No	Scoring Indicators	Split score	Total score
I.1.	• Bug discovery. • Bug prevention.	1x2	2
2.	1) Identify equivalence classes. 2) Design test cases	1x2	2
3.	• Checks if the bug has been addressed: The first objective in bug-fix testing is to check whether the bug-fixing has worked or not. • Finds other related bugs: It may be possible that the developer has fixed only the symptoms of the reported bugs without fixing the underlying bug • Tests to check the effect on other parts of the program: It may be possible that bug-fixing has unwanted consequences on other parts of a program . (Any two)	1x2	2
4.	Load runner is used for performance and load testing of a system. The tool is helpful for client/server application of various parameters like response time, no of users etc	1x2	2
5.	Debugging is the process of identification of symptoms of failures, tracing the bug, locating the errors that caused the bug, and correcting these errors. Consists of two processes : 1. Determining the exact nature of bug and location in program. 2. Fixing or repairing the error.	2	2
II. 1.	The software is basically a part of a system which it is being developed. Systems consist of software and hardware to make the program run. Testers work on the basis of bug model, which classifies the bugs based on the critically or the SDLC phase in which the testing is to be performed. **Model for software testing**  	1 2	

	Software and software model: Software is built after analysing the system in environment. Software model should design and code the software such that it is testable at every point.Thus the software to be tested may be modeled such that it is testable, avoiding unnecessary complexities. **Bug Model** – provides a perception of the kind of bugs expected. **Testing methodology and testing** : Based on the inputs from the software model and the bug model, testers can develop a testing methodology that incorporates both testing strategy and testing tactics.	3	6
2.	The steps to prepare the matrix are follows • Select and rank test factors • Identify system development phases • Identify risks associated with the system under development • Identify test phases • Identify risks associated with each test factors and its corresponding test phase • Plan test strategy for every risk identified.		6
3.	White-box testing is referred to as glass-box testing because everything that is required to implement the software is visible. NEED OF WHITE-BOX TESTING • White-box testing techniques are used for testing the module for initial stage testing. • White-box testing is complementary to black-box testing. Because different categories of bugs can be revealed by white-box testing, but not through black-box testing. • Errors which have come from the design phase will also be reflected in the code, so we must execute white-box test cases for verification of code (unit verification). • A logical path is not likely to be executed when, in fact, it may be executed on a regular basis; White-box testing explores these paths too. • A White-box testig technique helps to detect typographical errors.	1 1x5=5	6
4.	Regression testing is performed in the case of bug fixing or whenever there is a need to incorporate any new requirements in the software. After the first release whenever there is a need to fix the bugs reported or if there is any updation in the requirement, the bug is fixed or new requirement is incorporated by changing the code. This changed code requires to be fully tested again so as to ensure a) Bug fixing has been carried out successfully. b) Modification have not affected other unchanged modules. c) New requirement have been incorporated correctly this process is time consuming but increases the quality of software. Therefore regression testing helps to retain the quality of software the	4	6

	importance of regression testing is due to the following reasons • It validates the parts of software where changes occur • It validates the parts of software which may be affected by some changes, but otherwise unrelated. • It ensures proper functioning of the software. • It enhances the quality of software. Figure	2	
5.	GUIDELINES FOR AUTOMATED TESTING • consider building a tool instead of buiying one, if possible. • Test the tool on an application prototype. • Not all the tests should be automated. • Select the tools according to organizational needs. • Use proven test-script developement techniques;. • Automate the regression tests whenever feasible.	1x6	6
6.	Navigation testing is performed on various possible paths in web application. Design the test cases such that the following navigations are correctly executing: 1) Internal links 2) External links 3) Redirected links 4) Navigation for searching inside the web application. The errors much be checked for following i) The links should not be broken due to any reason ii) The redirected links should be with proper messages displayed to the user iii) Check that all possible navigation paths are active iv) Check that all possible navigation paths are relevant v) Check that navigations for back and forward buttons		6
7.	Debugging tools speeds up the process of debugging they help to track down, isolate and remove bugs from the program they can be used for the following. • Illustrate the dynamic nature of a program • Understand a program as well as to find and fix the bugs • Control the application using special facilities provided by the underlying OS.		6

4

III.	Software Testing methodology is the organization of software testing by means of which the test strategy and test tactics are achieved.  **Software Testing Strategy** **Testing strategy** is the planning of the whole testing process in to a well planned series of steps. Strategy provides a roadmap that includes very specific activities that must be performed by the test team in order to achieve a specific goal. **Test Factors** Test factor are risk factor or issues related to the system under development. Risk factor need to be selected and ranked according to a specific system under the development. The testing process should reduce these test factors to a prescribed level. **Test Phase** The phase of SDLC where testing will be performed. Testing strategy may be different for different models of SDLC. e.g. strategies will be different for waterfall and spiral models.	1 5 9	15
IV	The testing process is divided into a well defined sequence of steps termed as STLC STLC consists of following phases • Test planning • Test design • Test execution • Post excecution / test review Brief explanation of each phase	1½ x4 =6	

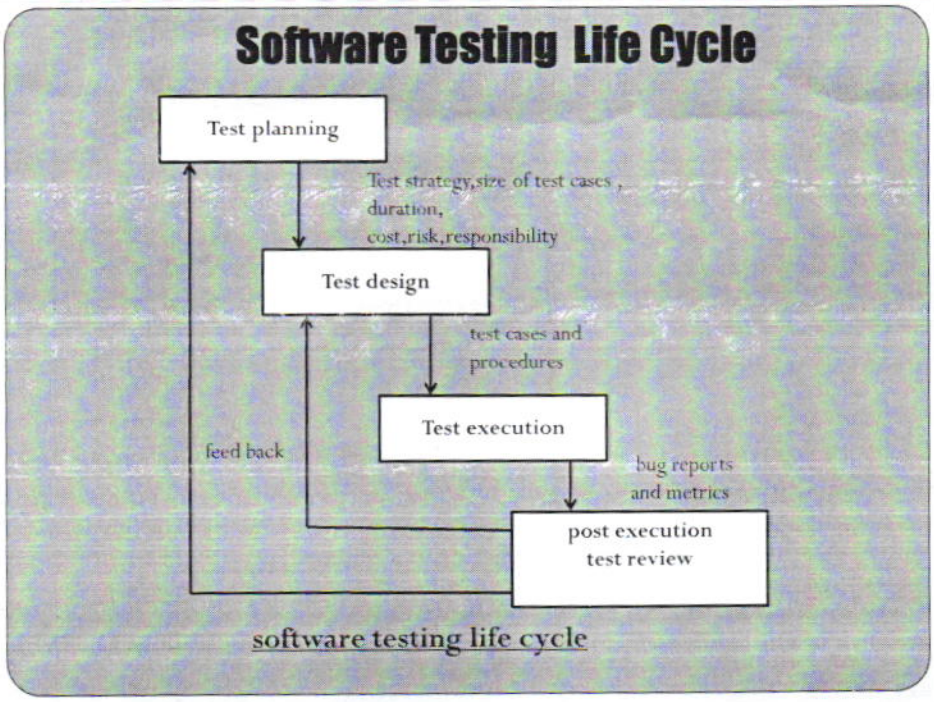


	Figure	3	9
	b) How to verify code: The points against which code is verified: • Check that every design specifications in HLD and LLD are coded uing traceability matrix. • Examine the code against a language specification check list. • Code verification can be done most efficiently by the developer. Specify the points against which the code can be verified : Misunderstood or incorrect arithmetic precedence, Mixed mode operation, Incorrect initialization etc Two kinds of techniques used to verify the coding : (a) Static testing and (b)Dynamic testing. Brief explanation of unit verification	3 1 2	6