18CSC206J SOFTWARE ENGINEERING AND PROJECT MANAGEMENT
UNIT-IV
SOFTWARE TESTING

1 MARKS

S.NO	QUESTIONS	LEVEL	CLO	PG. NO
1.	End result of Software Requirement Analysis is _____. **a. Functional and Behavioral** b. Architectural and Structural c. Usability and Reliability d. Algorithmic and Data Structure	1	4	R5 153
2.	Which Testing is performed first? a. Black box testing b. White box testing c. Dynamic testing **d. Static testing**	1	4	R5 194
3.	Verification and Validation uses _____. **a. Internal and External resources respectively.** b. Internal resources only. c. External resources only. d. External and Internal resources respectively.	1	4	R51 89
4.	Testing beyond normal operational capacity is _____. a. Load testing b. Performance testing **c. Stress testing** d. Dynamic testing	1	4	R1 479
5.	The expected results of the software is _____. **a. Only important in system testing** b. Only used in component testing c. Most useful when specified in advance d. Derived from the code	1	4	R1 712
6.	When an expected result is not specified in test case template then _____. a. We cannot run the test. b. It may be difficult to repeat the test. **c. It may be difficult to determine if the test has passed or failed.** d. We cannot automate the user inputs.	2	4	R5 173
7.	Test cases are created in which phase? **a. Test Specification** b. Test Planning c. Test Requirement d. Test Configuration	1	4	R1 449

8.	_____ refers to a different set of tasks ensures that the software that has been built is traceable to Customer Requirements. a. Verification b. Requirement engineering **c. Validation** d. None of the above	1	4	R1 450
9.	Which granularity level of testing checks the behavior of module cooperation? a) Unit Testing **b) Integration Testing** c) Acceptance Testing d) Regression Testing	1	4	R1 459
10.	Which testing is an integration testing approach that is commonly used when "shrink-wrapped" software products are being developed? a) Regression Testing b) Integration testing **c) Smoke testing** d) Validation testing	1	4	R1 463
11.	Which of the following is / are not a Iterative Model? a. RAD b. Incremental **c. V model** d. Spiral Model	1	4	R1 61
12.	Which is not true in case of Unit Testing? [REF 1, PAGE:466] **a. It decreases the software development speed.** b. It can't be expected to catch every error in a program. c. In this tester evaluates if individual units of source code are fit for use. d. It is usually conducted by the development team.	1	4	R1 466
13.	Focus Testing comes under _____. a. Performance Testing b. Acceptance Testing **c. Usability Testing** d. Component Testing	1	4	R1 540
14.	Difference between Retesting and Regression Testing is _____ **a. Retesting ensures the original fault has been removed where as regression testing looks for unexpected side-effects.** b. Retesting looks for unexpected side-effects where as regression testing ensures the original fault has been removed. c. Retesting is done after faults are fixed where as regression testing is done earlier d. Retesting is done by developers whereas regression testing is done by independent testers	1	4	R5 197

15.	First component of the DFD is _____ . **a. Process** b. Flow c. Entity d. Level	1	4	R1 187
16.	Minimum of four test data are available in _____ . **a. Boundary Value Analysis** b. Equivalence Class Partitioning c. Both A and B d. None of these.	1	4	R1 498
17.	Which coupling should be avoided in software? [REF 1, PAGE:289] a. Data coupling **b. Content Coupling** c. Control coupling d. Stamp coupling.	1	4	R1 289
18.	Cyclomatic Complexity cannot be applied in _____ . a. Re-engineering b. Risk Management c. Test Planning **d. Reverse engineering**	1	4	R1 488
19.	Data classification is done by which type of Decision Tree? a. Regression Tree b. Boosted Tree **c. Classification Tree** d. Bagging Tree	1	4	R1 715
20.	Which of the following is not a software testing generic characteristics? **a) Different testing techniques are appropriate at different points in time** b) Testing is conducted by the developer of the software or an independent test group c) Testing and debugging are different activities, but debugging must be accommodated in any testing strategy d) None of the mentioned	1	4	R5 335

4 MARKS:

1.	**What is software testing?** In most quality standards documents, software testing is divided into two parts: "validation" and "verification." While verification implies that the developed software is working as intended by checking the requirement specifications, design, source code, etc., in static mode, validation implies that the software has been validated to be working after running it and checking whether all functionality meets the requirements. Verification techniques are also known as static testing, since the source code is not run to do testing. Each work product including requirement specifications, design, and source code during software development is tested using static methods.	L2	CLO4

2. **How does software testing help in increasing quality of a software product?** L2 CLO4

It is a fact that the exact number of defects in a software product is difficult to find. At best it can be predicted using some defect estimation tools. It is also impossible to detect all defects in a software product. Nevertheless, finding and fixing critical bugs up to an acceptable limit as per expectations is important. If there are more defects in the product after the product enters production, then the project team will be in big trouble. The support costs for a bug ridden product will be too high. So, less than required testing is a certain call for rebuke from stakeholders. So the software testing helps in providing a quality software product which is bug free.

3. **What techniques are used for testing software?** L2 CLO4

Verification, Validation, Test Prioritization, Effort Estimation, Test Point Analysis, Defect Tracking

4. **List out the Problems with Traditional Development Model with neat diagram.** L2 CLO4

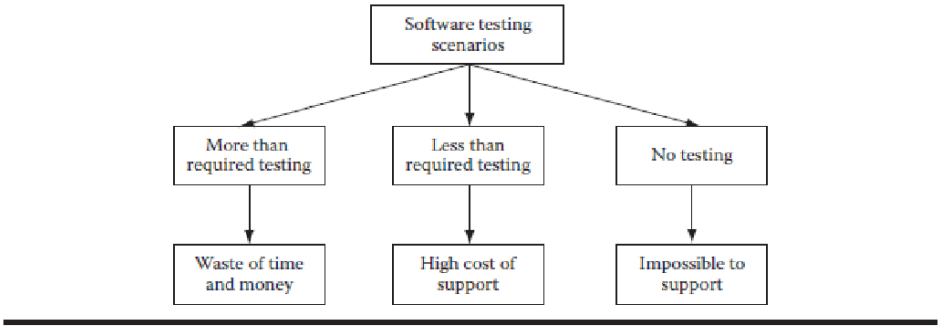

Figure 13.1 Software testing scenarios.

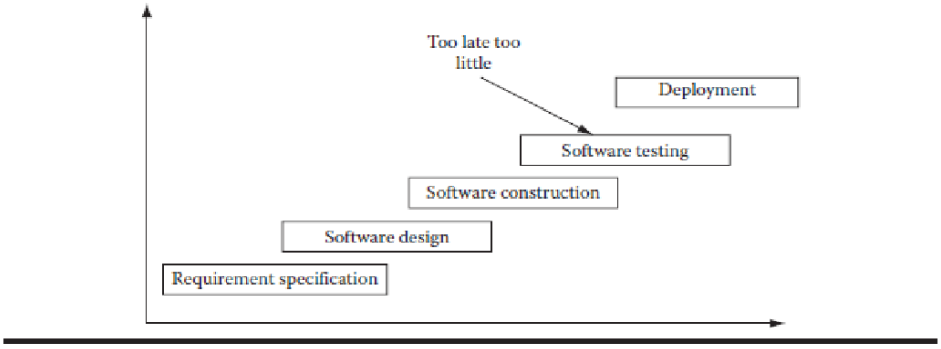

Figure 13.2 Traditional software development model (too little, too late testing).

5. **Describe about Verification and Validation.** L2 CLO4

In most quality standards documents, software testing is divided into two parts: "validation" and "verification." While verification implies that the developed software is working as intended by checking the requirement specifications, design, source code, etc., in static mode, validation implies that the software has been validated to be working after running it and checking whether all functionality meets the requirements.

Verification techniques are also known as static testing, since the source code is not run to do testing. Figure 13.3 shows that each work product including requirement specifications, design, and source code during software development is tested using static methods. The requirement specifications are reviewed for completeness, clarity, design ability, testability, etc. The software design is reviewed for robustness, security,

implementability, scalability, complexity, etc. The source code is reviewed for dead code, unused variables, faulty logic, constructs, etc.

Once the source code is ready to be run as a system, validation testing can be started. Validation testing is also known as dynamic testing as, in this case, the source code is actually run to determine that it is running per specifications. During validation, unit, integration, system, and finally user acceptance testing are performed. Unit testing is done to ensure each unit piece of source code is free from defects. Once unit testing is done, then this piece of code is integrated with the main source code build. But before integrating to the main build, it is strongly advisable to do local integration testing on the developer's own computer. Only when the source code runs smoothly and all integration tests pass, the source code should be integrated with the main build.

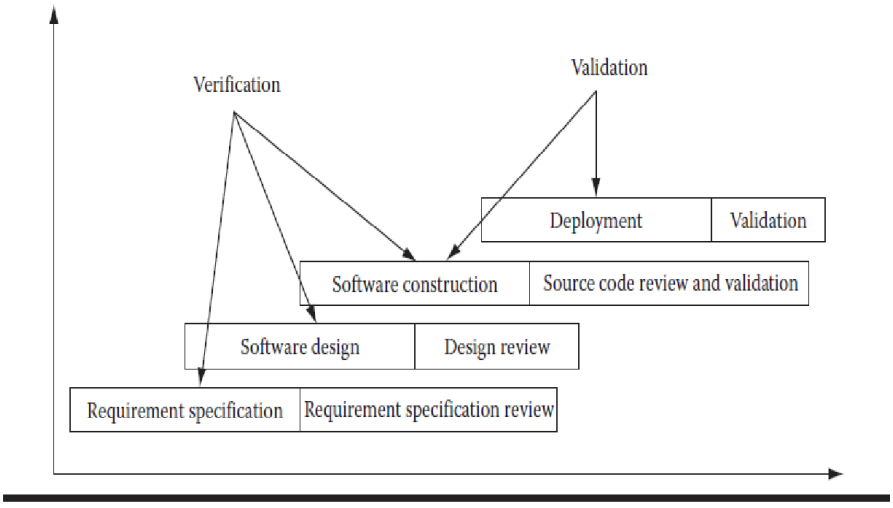

Figure 13.3 Software verification and validation.

6. **Write short notes on Risk management in Software testing.** L2 CLO4

The test manager should also do plan for all known risks that could impact the test project. If proper risk mitigation planning is not done, and a mishap occurs, then the test project schedule could be jeopardized, costs could escalate, and/or quality could go down.

Some of the risks that can have severe, adverse impact on a test project include an unrealistic schedule, resource unavailability, skill unavailability, frequent requirement changes, etc. Requirement changes pose a serious threat to testing effort, because for each requirement change, the whole test plan gets changed. The test team has to revise its schedule for additional work as well as to assess impact of the change on the test cases they have to recreate.

For test professional resources, a good alternative resource planning is required. The test manager should, in consultation with human resource manager, keep a line of test professionals who may join in case one is needed on his project.

For scheduling problems, the test manager has to ensure in advance that schedules do not get affected. He has to keep a buffer in the schedule for any eventuality.

To keep a tab on the project budget, the test manager has to ensure that the schedule is not unrealistic and also has to load his test engineers appropriately. If some test engineers are not loaded adequately, then project costs may go higher. For this

reason, if any test professionals do not have enough assignments on one project, they should be assigned work from other projects.

7. **State the importance of effort estimation?** L2 CLO4

For making scheduling, resource planning, and budget for a test project, the test manager should make a good effort estimate. Effort estimate should include information such as project size, productivity, and test strategy. While project size and test strategy information comes after consultation with the customer, the productivity figure comes from experience and knowledge of the team members of the project team.

The wideband Delphi technique uses brainstorming sessions to arrive at effort estimate figures after discussing the project details with the project team. This is a good technique because the people who will be assigned the project work will know their own productivity levels and can figure out the size of their assigned project tasks from their own experience. Initial estimates from each team member are then discussed with other team members in an open environment. Each person has his own estimate. These estimates are then unanimously condensed into final estimate figures for each project task.

Effort estimation is one area where no test manager can have a good grasp, at the initial stages of the project. This is because not many details are clear about the project. As the project unfolds, after executing some of its related tasks, things become clearer. At that stage, any test manager can comfortably give an effort estimate for the remaining project tasks.

8. **Explain about Test Automation.** L2 CLO4

Most testing tasks are done manually, as they are still difficult to automate. Wherever automation is possible, it can be evaluated. Care should also be taken not to do automation blindly. This is because the initial effort for automation is more than manual testing.

Testing tasks include requirements and design document review, test case scenario creation, test case creation, test case execution, test case management, and defect tracking.

Out of these tasks, test case execution and test case management are the only tasks for which good automation tools are available.

9. **Describe about Defect Tracking.** L2 CLO4

Defect tracking is one of the most important activities in a test project [13]. During defect tracking it is ensured that defects are logged and get fixed. All defects and their fixing are tracked carefully (Figure 13.6).

Defect count per hour per day is a common way of measuring performance of a test team. If the testing is done for an in-house software product, traditionally, it used to not be a performance evaluation measurement. What really counted was the number of defects found in production when the software product was deployed and used by end users. But it is too late a performance measurement. What if many of the test team members left before the product was deployed? In fact this is areality, given the high attrition rate (as much as 20% at many corporations) of software professionals. Once they are gone, there is no point in measuring the performance. Thus, a better

measurement would allow for more immediate results. This is achieved by measuring the defect count per hour per day. Then there is the case of outsourced test projects. If the contract is only for testing up to deployment and not afterward, then measurement does not make sense after the contract has ended.

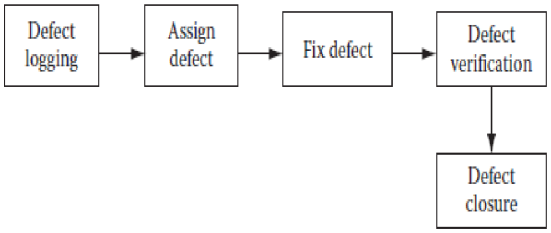

Figure 13.6 Defect life cycle.

A good defect tracking application should be deployed on a central server that is accessible to all test and development teams. Each defect should be logged in such a way that it could be understood by both development and testing teams.

10. **Write short notes on Software Testing in Iterative Model. [REF-5. PAGE: 197]** L2 CLO4

In an iterative model, each iteration is a short cycle. So the amount of testing in each iteration is also small. Thus, unlike in waterfall model, software testing has a lesser role in the iterative development life cycle.

Generally, software defects tend to increase with the size of software products. Since in iteration mode the software product is small, there will be fewer defects in the product. Although in reality, as the software product grows in size over many iterations, the number of defects per line of software code is bound to increase. In iterative development, regression testing is also a big issue. In each iteration, there will be a large number of regression test cases to run. As the product size increases with iterations, the set of regression test cases also increases. It becomes a liability after a while. Manually running all those regression tests takes a lot of time, which becomes a hindrance for the release schedule. In such cases, the best option is to go for automation of these regression test cases. Automated test cases take much less time (sometimes if the manual running of test cases was taking 5 days, after automation it took only 5 h) to run.

12 Marks:

1. Discuss in detail Test Strategy and Planning. L2 CLO4

2. Explain a case study on Software testing. L3 CLO4

3. (i) List out the Problems with Traditional Development Model with neat diagram L2 CLO4
   (ii) Describe about Verification and Validation. L2

4. (i) Write short notes on Risk management in Software testing. L2 CLO4
   (ii) State the importance of effort estimation. L1

5. (i) Explain about Test point analysis and its components. L2 CLO4
   L2

	(ii) Elaborate in detail the Test Automation.		
6.	(i) Describe about Defect Tracking. (ii)Write short notes on Software Testing in Iterative Model.	L2 L2	CLO4
7.	Elaborate in detail Test Project Monitoring and Control with neat sketch.	L2	CLO4
8.	Describe in detail the techniques used for testing software.	L2	CLO4