**UNIT-V**

**INTRODUCTION**

Testing is integrated into the endeavor of creating a given product or service; each phase and each type of testing has different characteristics and what is tested in each version could be different. Hence, testing satisfies this definition of a project fully.

Given that testing can be considered as a project on its own, it has to be planned, executed, tracked, and periodically reported on.

We will look at the test planning aspects in the next section. We will then look into the process that drives a testing project.

* **Test planning**
* Preparing a Test Plan

|  |
| --- |
| 1. What needs to be tested—the scope of testing, including clear identification of what will be tested and what will not be tested. 2. How the testing is going to be performed—breaking down the testing into small and manageable tasks and identifying the strategies to be used for carrying out the tasks. 3. What resources are needed for testing—computer as well as human resources. 4. The time lines by which the testing activities will be performed. |
|
|

1. Scope Management: Deciding Features to be Tested/Not Tested

Scope management pertains to specifying the scope of a project. For testing. scope management entails

1. Understanding what constitutes a release of a product;
2. Breaking down the release into features;
3. Prioritizing the features for testing;
4. Deciding which features will be tested and which will not be; and
5. Gathering details to prepare for estimation of resources for testing.

* Deciding Test Approach/Strategy

1. What type of testing would you use for testing the functionality?

1. What are the configurations or scenarios for testing the features?
2. What integration testing would you do to ensure these features work together?
3. What localization validations would be needed?
4. What "non-functional" tests would you need to do?

* Identifying Test Deliverables

.

1. The test plan itself (master test plan, and various other test plans for the project)
2. Test case design specifications
3. Test cases, including any automation that is specified in the plan
4. Test logs produced by running the tests
5. Test summary reports

* Testing Tasks: Size and Effort Estimation

Estimation happens broadly in three phases.

1. Size estimation
2. Effort estimation
3. Schedule estimation

We will cover size estimation and effort estimation in this sub-section and address schedule estimation in the next sub-section.

Size estimate quantifies the actual amount of testing that needs to be done. Several factors contribute to the size estimate of a testing project.

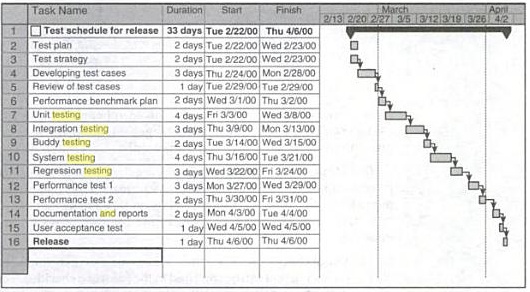
* Activity Breakdown and Scheduling

1. Identifying external and internal dependencies among the activities

1. Sequencing the activities, based on the expected duration as well as on the dependencies
2. Identifying the time required for each of the WBS activities, taking into account the above two factors
3. Monitoring the progress in terms of time and effort
4. Rebalancing schedules and resources as necessary

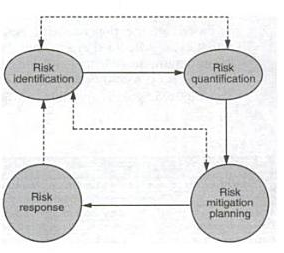
Activity Some of the common external dependencies are

1. Availability of the product from developers;
2. Hiring;
3. Training;
4. Acquisition of hardware/software required for training; and
5. Availability of translated message files for testing.

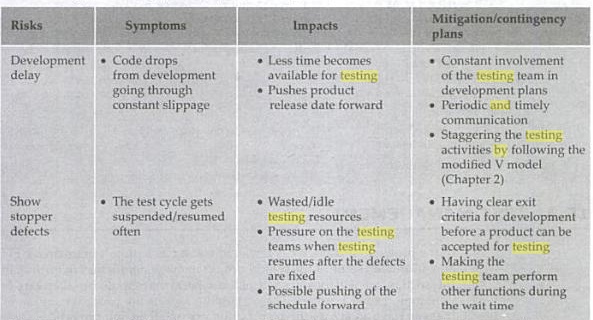


* **Risk Management**

Just like every project, testing projects also face risks. Risks are events that could potentially affect a project's outcome.



Use of checklists Over time, an organization may find new gleanings on testing that can be captured in the form of a checklist. For example, if during installation testing, it is found that a particular step of the installation has repeatedly given problems, then the checklist can have an explicit line item to check that particular problem.



* **Test management**

Test management most commonly refers to the activity of managing the computer software testing process.

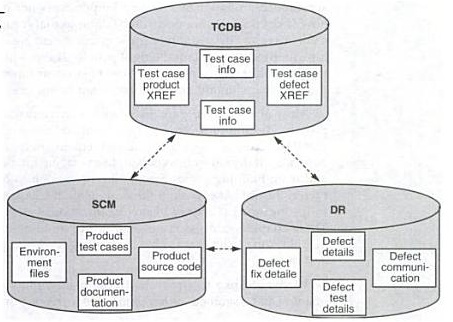
A test management tool is [software](http://en.wikipedia.org/wiki/Software) used to manage [tests](http://en.wikipedia.org/wiki/Test_case) (automated or manual) that have been previously specified by a test procedure. It is often associated with [automation](http://en.wikipedia.org/wiki/Automation) software.

Test management tools often include [requirement](http://en.wikipedia.org/wiki/Requirement) and/or [specification](http://en.wikipedia.org/wiki/Specification_(technical_standard)) management modules that allows automatic generation of the [requirement test matrix](http://en.wikipedia.org/wiki/Traceability_matrix) (RTM), which is one of the main metrics to indicate functional coverage of a [system under test](http://en.wikipedia.org/wiki/System_under_test) (SUT).

* **Test infrastructure management**

1. Changes to test files are made in a controlled fashion and only with proper approvals.

1. Changes made by one test engineer are not accidentally lost or overwritten by other changes.
2. Each change produces a distinct version of the file that is recreatable at any point of time.
3. At any point of time, everyone gets access to only the most recent version of the test files (except in exceptional cases).



Similarly, in order to decide which tests to run for a given regression run,

1. The defects recently fixed can be obtained from the defect repository and tests for these can be obtained from the TCDB and included in the regression tests.
2. The list of files changed since the last regression run can be obtained from the SCM repository and the corresponding test files traced from the TCDB.
3. The set of tests not run recently can be obtained from the TCDB and these can become potential candidates to be run at certain frequencies

* Test People Management

People management is an integral part of any project management. Often, it is a difficult chasm for engineers-turned-managers to cross.

As an individual contributor, a person relies only on his or her own skills to accomplish an assigned activity; the person is not necessarily trained on how to document what needs to be done so that it can be accomplished by someone else

Most of the above gaps in people management apply to all types of projects. Testing projects present several additional challenges.

We believe that the success of a testing organization (or an individual in a testing career) depends vitally on judicious people management skills.

* Test execution

A test plan combines all the points discussed above into a single document that acts as an anchor point for the entire testing project.

### Traceability Matrix

### A requirements traceability matrix is a document that traces and maps user requirements [requirement Ids from requirement specification document] with the test case ids. Purpose is to make sure that all the requirements are covered in test cases so that while testing no functionality can be missed.

* This document is prepared to make the clients satisfy that the coverage done is complete as end to end, this document consists of Requirement/Base line doc Ref No., Test case/Condition, and Defects/Bug id. Using this document the person can track the Requirement based on the Defect id
* **Baseline testing**
* It is one of the type of [non-functional testing](http://istqbexamcertification.com/what-is-non-functional-testing-testing-of-software-product-characteristics/).
* It refers to the validation of documents and specifications on which test cases would be designed. The requirement specification validation is baseline testing.
* Generally a baseline is defined as a line that forms the base for any construction or for measurement, comparisons or calculations.
* Baseline testing also helps a great deal in solving most of the problems that are discovered. A majority of the issues are solved through baseline testing.
* The Test Execution Process
* The following outlines what I consider a good test execution process.
* 1. Based on an overall quality risk management strategy, select a subset of test suites from the test
* Cohort for this test cycle.
* 2. Assign the test cases in each test suite to testers for execution.
* 3. Execute tests, report bugs, and capture test status continuously.
* 4. Resolve blocking issues as they arise.
* 5. Report status, adjust assignments, and reconsider plans and priorities daily.
* 6. Manage the test cycle end game, eliminating unrealizable tests in reverse-priority (lowest first,
* highest last) order.

### Test Reporting

Testing requires constant communication between the test team and other teams (like the development team).

Test reporting is a means of achieving this communication. There are two types of reports or communication that are required: test incident reports and test summary reports (also called test completion reports).

#### Test incident report

A test incident report is a communication that happens through the testing cycle as and when defects are encountered. Earlier, we described the defect repository.

A test incident report is nothing but an entry made in the defect repository. Each defect has a unique ID and this is used to identify the incident.

The high impact test incidents (defects) are highlighted in the test summary report.

### Best Practices

Best practices in testing can be classified into three categories.

1. Process related
2. People related
3. Technology related

* **Process related best practices**

Ensuring people friendly processes makes for process friendly people

* **People related best practices**

The key to successful management is to ensure that the testing and development terms gel well. this gelling can be enhanced by creating a sense of ownership in the overarching product goals.

* **Technology released best practices**

Twenty first century tools with nineteenth century processes can only lead to nineteenth century productivity.

* **Project Metrics**

Project metrics provide a means of measuring where we stand on the project and how close we are to achieving project objectives.

They give the project team insights that can help them guide the project toward these objectives if current status and trends indicate that the project team might not achieve these objectives within project constraints

Effectiveness project metrics measure the extent to which the project is on track to achieve desired results.

Efficiency project metrics measure the extent to which a project achieves those desired results in an economical fashion. Elegance project metrics measure the extent to which a project effectively and efficiently achieves those results in a graceful, well-executed fashion.

Project metrics are the most commonly used test metrics. Tables and graphs showing status and trends in terms of bugs and test cases are used on most projects with any

Developing Good Project Metrics

1. Define test- and quality-related objectives for the project.
2. Consider questions about the effectiveness, efficiency, and elegance with which the project is achieving those objectives.
3. Devise measurable metrics, either direct or surrogate, for each effectiveness, efficiency, and elegance question.
4. Determine realistic goals for each metric, such that the quality and testing for the project will be adequate upon release.
5. Monitor progress towards those goals, determining project status and making test and project control decisions as needed to optimize project outcomes.

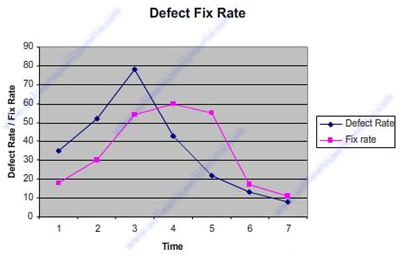
### Progress Metrics

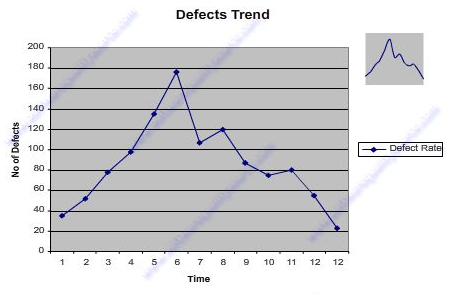
Any project needs to be tracked from two angles. One, how well the project is doing with respect to effort and schedule.

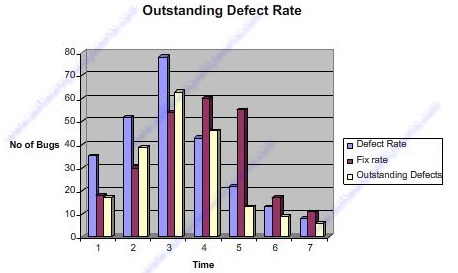
The other equally important angle is to find out how well the product is meeting the quality requirements for the release.

There is no point in producing a release on time and within the effort estimate but with a lot of defects, causing the product to be unusable. One of the main objectives of testing is to find as many defects as possible before any customer finds them.

The number of defects that are found in the product is one of the main indicators of quality. Hence in this section, we will look at progress metrics that reflect the defects (and hence the quality) of a product.







### Productivity Metrics

Productivity metrics combine several measurements and parameters with effort spent on the product. They help in finding out the capability of the team as well as for other purposes, such as

1. Estimating for the new release.
2. Finding out how well the team is progressing, understanding the reasons for (both positive and negative) variations in results.
3. Estimating the number of defects that can be found.
4. Estimating release date and quality.
5. Estimating the cost involved in the release.

### Release Metrics

The decision to release a product would need to consider several perspectives and several metrics. All the metrics that were discussed in the previous sections need to be considered in totality for making the release decision.

The purpose of this section is to provide certain guidelines that will assist in making this decision. These are only set of guidelines and the exact number and nature of criteria can vary from release to release, product to product, and organization to organization.

| Some perspectives and sample guidelines for release analysis. | | |
| --- | --- | --- |
| Metric | Perspectives to be considered | Guidelines |
| Test cases executed | Execution % Pass % | * All 100% of test cases to be executed * Test cases passed should be minimum 98% |
| Effort distribution | Adequate effort has been spent on all phases | * 15–20% effort spent each on requirements, design, and testing phases |
| Defect find rate | Defect trend | * Defect arrival trend showing bell curve * Incoming defects close to zero in the last week |
| Defect fix rate | Defect fix trend | * Defect fixing trend matching arrival trend |
| Outstanding defects trend | Outstanding defects | * Outstanding defects trend showing “downward” trend * Close to zero outstanding defects in the last few weeks prior to release |
| Priority outstanding defects trend | High-priority defects | * Close to zero high-priority defects in the last few weeks prior to release |
| Weighted defects trend | High-priority defects as well as high number of low-priority defects | * Weighted defects trend showing “bell curve” * Close to zero weighted defects in the last few weeks prior to release |
| Defect density and defect removal rate | Defects/KLOC Defect removal % | * Defects/KLOC less than 7 * Defects/KLOC less than last release * Defect removal percentage is 50% more * Defect removal percentage better than last release |
| Age analysis of outstanding defects | Age of defects | * Age of defects showing downward trend |
| Introduced and reopened defects | Quality of defect fix Same defects reappearing again | * Combined number of outstanding and reopened defects showing downward trend * Introduced and reopened defects are less than 5% of defect arrival rate |
| Defects per 100 hours of testing | Whether defect arrival is proportional to effort spent | * Defects per 100 hours of testing should be less than 5 * Defects per 100 hours of testing trend showing downward trend |
| Test cases executed per 100 hours of testing | Whether improved quality in product allowing more test cases being executed Whether test cases executed is proportional to effort spent | * Test cases executed showing an upward trend |
| Test phase effectiveness | Defects found in each of the test phases | * Very low percentage of defects found in system and acceptance test phase (say less than 12%) * A distribution of defects and reduction in defects % compared to next test phase * A distribution of UT = 50%, CT = 30%, IT = 15% and ST = 5% would be ideal |
| Closed defects distribution | Whether good proportion of defects found by testing are fixed | * At least 70% of closed defects are fixed * Non-reproducible defects are less than 5% * Defects moved to next release should be less than 10% |

**ONE MARK QUESTIONS**

1. --------------pertains to specifying the scope of a project
2. **Scope management** b. test planning c. test reporting d. none
3. --------------- quantifies the actual amount of testing that needs to be done
4. **Size estimate** b. effort estimation c. schedule estimate d. none
5. ---------------consists of indentifying the possible risks that may hit a project
6. **Risk identification** b. quantifying risks c. checklist d. none
7. -----------------deals with expressing the risk in numerical terms
8. **Risk quantification** b. risk identification c. risk mitigation d. none
9. The success of testing depends a lot on knowing what the correct expected behavior of the product under test is------------------
10. **Unclear requirements** b. risk quantification c. use is checklist d. none
11. -------------the schedule of the testing team depends significantly on the schedules of the development team
12. **Schedule dependence** b. unclear requirements c. risk quantification d. none
13. SCM stands for-----------------
14. **Software Configuration management** b. system configuration management c. system content management d. none
15. -------------------- is an integral part of any project management
16. **People management** b. system management c. test management d. none
17. --------------------is a means of achieving the communication
18. **Test reporting** b. test planning c. test management d. none
19. ------------------ is the actual time that is spent on a particular activity or a phase
20. **Effort** b. metrics c. report d. management
21. An approach involved in getting the granular detail is called--------------
22. **Data drilling** b. measurements c. effort d. none
23. The purpose of development is to fix defects as soon as they arrive is-------------
24. **Defect fix rate** b. defect find rate c. outstanding rate d. none
25. -------------is several measurements and parameters with effort spent on the product
26. **Productivity metrics** b. product metrics c. project metrics d. none
27. The product is ready for release is called----------
28. **Release metrics** b. product metrics c. report metrics d. none
29. TCDB stands for--------------
30. **Test Case Database** b. test condition database c. test code database d. none
31. Test case specification becomes the basis for preparing individual-----------------
32. **Test cases** b. test report c. test plan d. all
33. New defects that get uncovered in the current run of the test is-----------
34. **Executing test cases**  b. test planning c. test reporting d. none
35. The success of testing depends a lot on knowing what the correct expected behavior of the product under test is------------------
36. **Unclear requirements** b. schedule dependence c. risk quantification d. none
37. Productivity refers to the speed at which the various activities of testing can be carried out is------------
38. **Productivity data** b. reuse opportunities c. robustness d. none
39. Special requirements for running machine intensive test such as-----------------
40. Load test b. performance test c. system test d. **a and b**

**ANSWER THE FOLLOWING QUESTIONS**

5 MARKS

1. What is test planning?
2. What is test reporting?
3. What is project metrics?
4. What is progress metrics?
5. What is test process?

8 MARKS

1. Explain test management?
2. Explain process and reporting?
3. Describe project metrics?
4. Brief explain productivity metrics?
5. Explain release metrics?