Test Management

Test Management is a process where testing activities are managed to ensure high-quality and high-end testing of software applications.

It is a software process that manages the start to the end of all software testing activities. This management process provides planning, controlling, tracking, and monitoring facilities throughout the whole group cycle, these process includes several activities like test case design and test execution, test planning, etc.

# Role of Test Manager

Typical test leader tasks may include:

* Coordinating and planning with project managers and others.
* Write and update test plan(s).
* Estimating the time and effort and converting to the cost of testing and acquiring the right resources.
* Monitoring and control of all the specification and execution.
* Taking the required action, including adapting the planning, based on test results and progress, and any action necessary to compensate for problems or delays.
* Putting in place suitable metrics for measuring test progress and evaluating the quality of the testing delivered and the product.
* Deciding what should be automated
* Selecting tools to support testing and ensuring any tool training requirements are met.
* At the end of the project, writing a test summary report based on the information gathered during testing.

# TASKS UNDERTAKEN BY A TESTER

The tasks typically undertaken by a tester may include:

1) Reviewing and contributing to the development of test plans.

2) Analysing, reviewing and assessing user requirements, specifications and models for testability.

3) Creating test design

4) Preparing and acquiring/copying/creating test data.

5) Implementing tests on all test levels, executing and logging the tests, evaluating the results and documenting the deviations from expected results as defects.

6) Using test administration or management and test monitoring tools as required.

7) Automating tests that may be supported by a developer or a test automation expert.

8) Where required, running the tests and measuring the performance of components and systems.

9) Reviewing tests developed by other testers.

# Test Plan

Test planning is the most important activity undertaken by a test leader in any test project. It ensures that there is initially a list of tasks and milestones in a baseline plan to track progress against, as well as defining the shape and size of the test effort.

|  |  |  |
| --- | --- | --- |
| Section No. | Heading | Details |
| 1 | Test plan identifier | A unique identifying reference such as  Doc ref XYZ v2? |
| 2 | Introduction | A brief introduction to the document and the project for which it has been produced |
| 3 | Test items | # A test item is a software item that is the object of testing  # A software item is one or more items of source code, object code, job control code, or control data  # This section should contain any documentation references, e.g. design documents |
| 4 | Features to be tested | # A feature is a distinguishing characteristic of a software item (e.g. performance, portability, or functionality)  # Identify all software features and combinations of features and the associated test design specification |
| 5 | Features not to be tested | Identify all software features and significant combinations and state the reasons for not including them |
| 6 | Approach | Details the overall approach to testing; this could include a detailed process definition, or could refer to other documentation where the detail is documented, i.e. a test strategy |
| 7 | Item pass / fail criteria | Used to determine whether a software item has passed or failed its test  How many critical requirements have been met by the application?  How many outstanding defects are there? |
| 8 | Suspension and resumption requirements | Suspension requirements define criteria for stopping part or all of the testing activity Resumption requirements specify the requirements to resume testing  Smoke tests –  Unit Test Case Result |
| 9 | Test deliverables | The documents that testing will deliver, e.g. from IEEE 829 documents such as: # Test plans (for each test level)  # Test specifications (design, case and procedure)  # Test summary reports |
| 10 | Testing tasks | All tasks for planning and executing the testing, including the inter-task dependencies (Ex:cannot start testing without test environment) |
| 11 | Environmental needs | Definition of all environmental requirements such as hardware, software, PCs, desks, stationery, etc. |
| 12 | Responsibilities | Identifies the roles and tasks to be used in the test project and who will own them |
| 13 | Staffing and training needs | Identifies any actual staffing requirements and any specific skills and training requirements, e.g. automation |
| 14 | Schedule | Document delivery dates and key milestones |
| 15 | Risks and contingencies | High-level project risks and assumptions and a contingency plan for each risk |
| 16 | Approvals | Identifies all approvers of the document, their titles and the date of signature |

Factors influencing the test planning process:

1. Risks

2. Constraints

3. Resources

4. Criticality

5. Scope of testing

6. Testing Objectives

7. Testability.

# Entry and Exit criteria

Entry criteria are used to determine when a given test activity can start. This could include the beginning of a level of testing, when test design and/or when test execution is ready to start.

Examples of some typical entry criteria to test execution include:

1) Test environment available and ready for use (it functions).

2) Test tools installed in the environment are ready for use.

3) Testable code is available.

4) All test data is available and correct.

5) All test design activity has completed.

6)Availability of testable requirements, user stories and models.

7)Availability of staffs.

## Exit criteria

Exit criteria are used to determine when a given test activity has been completed or when it should stop. Exit criteria can be defined for all of the test activities, such as planning, specification and execution as a whole, or to a specific test level for test specification as well as execution.

Exit criteria should be included in the relevant test plans.

Some typical exit criteria are:

1) All tests planned have been run.

2) A certain level of requirements coverage has been achieved.

3) No high-priority or severe defects are left outstanding.

4) All high-risk areas have been fully tested, with only minor residual risks left outstanding.

5) Cost—when the budget has been spent.

6) The schedule has been achieved, e.g. the release date has been reached and the product has to go live.

7)Quality – The status of the important quality characteristics for the system for example reliability, performance efficiency usability, security etc.

8)Risk – the undesirable outcomes that could result from shipping too early.

# Types of Risks

In a project a test leader will use risk in two different ways:   
  
1) Project risks   
  
2) Product risks  
  
In both type of risks the calculation of the risk will be as per the following formula  
  
**Level of risk = (Probability of the risk occurring) × (Impact if it did happen)**

Probability : chance of the risk being realized

Impact - what will happen if the risk occurs

Project risks include:

**Supplier issues:**

a) Failure of a third party to deliver on time or at all.

b) Contractual issues, such as meeting acceptance criteria.

**Organizational factors:**

1) Skills, training and staff shortages.

2) Personal issues.

3) Political issues, such as problems that stop testers communicating their needs and test results.

1. Delays may occur in delivery, task completion or satisfaction of exit criteria or definition of done.

Inaccurate estimates, reallocation of funds to higher priority projects or general cost cutting across organization.

Late changes may result in substantial rework.

A Risk Analysis template:

**Project Risk**

**Technical issues:**

a) Problems in defining the right requirements.

b) Test environment not ready on time.

C) Low quality of the design, code, configuration data, test data and tests.

d)Poor defect management.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product Risk | Likelihood | Impact | Risk Priority # | Mitigation |
| Risk Category1 |  |  |  |  |
| Risk 1 |  |  |  |  |
| Risk 2 |  |  |  |  |
| Risk n |  |  |  |  |

# Test Effort Estimation

Test Effort is influenced by the type of project, scope, risks, test objective .

There are two techniques for estimation covered by the ISTQB Foundation Syllabus.

1. The first involves analyzing metrics from past projects and from industry data. (Metrics-based )

2. The second involves consulting the people who will do the work and other people with expertise on the tasks to be done. (Expert -based)

### Metrics-Based Approach of Test Estimation

Metrics based approach relies upon data collected from previous or similar projects.

This kind of data can include the following:

1) The number of test conditions.

2) The number of test cases written.

3) The number of test cases executed.

4) The time taken to develop test cases.

5) The time taken to run test cases.

6) The number of defects found.

7) The number of environment outages and how long on average each one lasted.

8)Testers to Developers ration(Top down approach)

### Expert-Based Approach of Test Estimation

This approach uses the experience of owners of the relevant tasks or experts to derive an estimate (this is also known as the Wide Band Delphi approach).

In this context ‘experts’ could be:

1) Business experts.  
2) Test process consultants.  
3) Developers.  
4) Technical architects.  
5) Analysts and designers.  
6) Anyone with knowledge of the application to be tested or the tasks involved in the process.

# TEST PROGRESS MONITORING AND CONTROL

The purpose of test monitoring is to give feedback and visibility about test activities.

Having developed the plans, defined the test strategies and approaches and estimated the work to be done, we must now track our testing work as we carry it out. Test monitoring can serve various purposes during the project, including the following:

• **Give the test team and the test manager feedback on how the testing work is going**, allowing opportunities to guide and improve the testing and the project.

• **Provide the project team with visibility about the test results.**

• **Measure the status of the testing**, test coverage and test items against the exit criteria to determine whether the test work is done.

• **Gather data for use in estimating future test efforts.**

## Common test metrics include:

1. Percentage of work done in test case preparation (or percentage of planned test cases prepared).
2. Percentage of work done in test environment preparation.
3. Test case execution (e.g. number of test cases run/not run, and test cases passed/failed).
4. Defect information (e.g. defect density, defects found and fixed, failure rate, and retest results).
5. Test coverage of requirements, risks or code.
6. Dates of test milestones.
7. Testing costs, including the cost compared to the benefit of finding the next defect or to run the next test.
8. Status of Testing.

# Test Reporting

Test progress monitoring is about gathering detailed test data; reporting test status is about effectively communicating our findings to other project stake-holders.

There are two types of reports

**Test Progress report** and **Test Summary report.**

Test Progress report will have:

* The Current status of the test activities and progress against the test plan.
* Factors that are currently impending the progress of the testing.
* The testing planned for the next period

Test Summary report will have:

* Summary
* Any deviations
* Summary of activities
* Approvals

A good summary report will have:

1. Describe scope of testing
2. Test environment details
3. Types of testing performed
4. Lessons learned
5. Status of exit criteria

# Issue management

* Issue management refers to the process of identifying and resolving the issues.
* Issues can be bugs, performance issues, UI issues, compatibility issues or suggested improvements.
* Issue management involves identifying, analyzing, prioritizing, and tracking issues from the point they’re identified till they are eventually resolved.
* [Issue tracking](https://kissflow.com/issue-tracking/what-is-issue-tracking-software/) may also involve steps taken to prevent the recurrence of such issues in the future.

## Issue management best practices

* Record issues
* Detail issues – specify type and priority of issues
* Mention issue severity
* Maintain issue log
* Monitor issue resolution