**Chapter 5: TEST MANAGEMENT**

1. **INTRODUCTION**

* This chapter provides a generic overview of how testing is organized and how testing is managed within organizations.
* A generic view of testing will, inevitably, not match the way testing is organized in specific organizations.
* The issues addressed are nevertheless important for any organization and need to be considered by all.
* We will start by looking at how testing and risk fit together, as well as providing detailed coverage of test planning and the control of testing, and we will identify how independence assists the test process.
* One very important area in managing the test process is the understanding of the different roles and tasks associated with the testing role such as the test manager and the tester.
* We cannot, in one chapter, provide all the knowledge required to enable the reader to become a practicing test manager, but we do aim to provide the background information necessary for a reader to understand the various facets of the test management role.

1. **RISK AND TESTING**

* **Risk:** A factor that could result in future negative consequences usually expressed as impact and likelihood.
* Level of Risk = Probability of the risk occurring x impact if it did happen.
* Identifies the following:
  + Will determine the test techniques to be employed, and/or the extent of testing to be carried out; for example, the Motor Industry Software Reliability Association (MISRA) defines which test techniques should be used for each level of risk: the higher the risk, the higher the coverage required from test techniques.
  + Will determine the levels and types of testing to be performed, such as security testing or accessibility testing.
  + Will determine the extent of testing to be carried out. (For example, what the depth of test coverage should be)
  + Prioritize testing in an attempt to find the critical defects as early as possible; for example, by identifying the area’s most likely to have defects (the most complex) the testing can be focused on these areas.
  + Will determine any non-test activities that could be employed to reduce risk. (For example, to provide training to inexperienced designers.)

1. **Project Risks**
   1. Project Issues

* Delays in delivery
* Inaccurate estimates
* Late changes
  1. Organizational Issues
* The skills and training of our staff may be inadequate.
* Personal issues between staff impacting progress.
* Users, business staff of subject matter experts may be unavailable when needed.
  1. Political Issues
* Testers may not communicate their needs and/or test results adequately.
* Developers and/or testers may fail to follow up on information found in testing and reviews, for example not following up on process improvements identified.
* There may be an improper attitude to or understanding of the value of testing.
  1. Supplier Issues
* Failure of a third party to deliver on time or at all.
* Contractual issues, such as meeting acceptance criteria.
  1. Technical Issues
* Problems in defining the right requirements.
* The extent that requirements can be met given existing project constraints.
* Test environment not ready on time.
* Late data conversion, migration, planning and development, and testing data conversion/migration tools.
* Weakness in the development process that impacts the quality of work products.
* Poor defect management resulting in an increase in technical debt.
* Low quality of the design, code, configuration data, test data and tests.

1. **Product Risks (Examples)**

* Failure-prone software delivered – not able to perform as intended according to the specification and/or the user requirements.
* System architecture may not adequately support non-function requirement(s).
* A particular computation may be performed incorrectly in certain circumstances.
* A loop structure may be coded incorrectly.
* Feedback from users indicates that the product may not meet expectations.
* The potential that a defect in the software/hardware could cause harm to an individual or company.
* Poor data integrity and quality. (Data migration issues, data conversion problems, data transport problems, violation of data standards.
* Software that does not perform its intended functions.

1. **TEST ORGANIZATION**

**A diagram of a test

Description automatically generated**

1. **Test Organization and Independence**

* Independent testers are important because it is genuinely hard for authors to identify their own errors, but it is easier for others to see them.

A white and black text on a white background

Description automatically generated

1. **Tasks of a Manager and Tester**

* Test tasks are normally performed by testers but can also be from non-testers such as project manager, quality manager, developer, business and domain experts, infrastructure personnel or IT operations.
* Tasks of test manager
  + Coordinating or developing the test policy and test strategy for the organization.
  + Planning the test activities by considering the context and understanding the test objectives and risks. This may include selecting test approaches, estimating test time, effort, and cost, acquiring resources, defining test levels and test cycles, and planning defect management.
  + Writing and updating test plan(s).
  + Coordinating the test plan(s) with project managers, product owners and others.
  + Sharing test perspectives with other project activities, such as the code integration planning.
  + Initiating the analysis, design, implementation, and execution of tests, monitoring test progress and results, and checking the status of execution criteria (or definition of ‘done’).
  + Preparing and delivering test progress reports and test summary reports based on the information gathered.
  + Adapting planning based on test results and progress; for example, if more defects than planned are found, this will impact the time taken to complete testing and so action will need to be taken to realign the plan.
  + Supporting the setting up of the defect management system and adequate configuration management of test ware.
  + Introducing suitable metrics for measuring test progress and evaluating the quality of the testing and the product.
  + Supporting the selection and implementation of tools to support the test process, including budget, and the allocation of time for the effort required to build and support tools.
  + Deciding about the implementation of test environment(s).
  + Promoting and advocating the tester, the test team, and the test profession within the organization.
  + Developing the skills and careers of testers through training plans, performance evaluations, coaching and so on.
* Tasks of tester
  + Reviewing and contributing to test plans.
  + Analyzing, reviewing, and assessing user requirements, user acceptance criteria, specifications and models for testability. stories and
  + Creating test specifications from the test basis; for example, test conditions, and the traceability between test cases, test conditions and the test basis.
  + Setting up the test environment (often coordinating with system administration and network management). In some organizations the setting up and management of the test environment could be centrally controlled; in this situation a tester would directly liaise with the environment management to ensure that the test environment is delivered on time and to specification.
  + Designing and implementing test cases and test procedures.
  + Preparing and acquiring/copying/creating test data.
  + Executing tests on all test levels, logging the tests, evaluating the results, and documenting the deviations from expected results as defects.
  + Using test administration, or management and test monitoring tools as required.
  + Automating tests (may be supported by a developer or a test automation expert).
  + Evaluating non-functional characteristics reliability and usability.
  + Reviewing tests developed by other testers.

1. **TEST STRATEGY AND TEST APPROACHES**

* Testing either in a project or company wide
  + Developed early in the life cycle, which is known as preventative – in this approach the test design process is initiated as early as possible in the life cycle to stop defects being built into the final solution.
  + Left until just before the start of test execution, which is known as reactive – this is where testing is the last development stage and is not started until after design and coding have been completed (sometimes it is identified as the waterfall approach, i.e., all development stages are sequential, the next not starting until the previous one has nearly finished)
* Strategies that can be employed
  + Analytical strategies rely on the analysis of some factors such as risk-based testing, where testing is directed to areas of greatest risk (see earlier in this chapter for an overview of risk-based testing).
  + Model-based strategies base tests on a model such as statistical information about failure rates (such as reliability growth models) or usage models (such as operational profiles).
  + Methodical strategies rely on the systematic use of some predefined tests or test conditions such as failure based (including error guessing and fault attacks), checklist based and quality-characteristic based.
  + Process-compliant (or standard-compliant) strategies adhere to the processes developed for use with standards (see ISO/IEC/IEEE 29119-2 or MISRA) and various Agile or traditional waterfall approaches.
  + Reactive (or dynamic and heuristic) strategies, such as exploratory testing where testing is more reactive to events than pre-planned, and where execution and evaluation are concurrent tasks.
  + Directed (or consultative) strategies, such as those where test coverage is driven primarily by the advice and guidance of technology and/or business domain experts outside or within the test team.
  + Regression-averse strategies are designed with the desire to avoid regression of existing capabilities such as those that include reuse of existing test material, extensive automation of functional regression tests and standard test suites.

1. **TEST PLANNING AND ESTIMATION**
2. **Test Planning**

* 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.
* Test planning is used in development and implementation projects (sometimes called ‘greenfield’) as well as maintenance (change and fix) activities.
* The main document produced in test planning is often called a master test plan or a project test plan. This document defines at a high level the test activities being planned. It is normally produced during the early phases of the project (e.g. initiation) and updated as required via change control as the project develops. It will provide sufficient information to enable a test project to be established (bearing in mind that at this point in a project little more than requirements may be available from which to plan).

A diagram of a test plan

Description automatically generated

1. **Test-Planning Activities**

* During test planning various activities for an entire system or a part of a system have to be undertaken by those working on the plan. They include:
  + Working with the project manager and subject matter experts to determine the scope and the risks that need to be tested. Also identifying and agreeing the objectives of the testing, be they time, quality or cost focused, or a mixture of all three. The objectives will enable the test project to know when it has finished – has time or money run out, or has the right level of quality been met?
  + Understanding what delivery model is to be used (waterfall, iterative, Agile, etc.) and defining the overall approach of testing (sometimes called the test strategy) based on this, ensuring that the test levels and entry and exit criteria are defined.
  + Liaising with the project manager and making sure that the testing activities have been included within the software life cycle activities such as:
    - design – the development of the software design.
    - development – the building of the code.
    - implementation environment. – the activities surrounding implementation into a live
  + Working with the project to decide what needs to be tested, what roles are involved and who will perform the test activities, planning when and how the test activities should be done, deciding how the test results will be evaluated, and defining when to stop testing (exit criteria).
  + Building a plan that identifies when and who will undertake the test analysis and design activities. In addition to the analysis and design activities, test planning should also document the schedule for test implementation, execution, and evaluation. The plan can either be sequential; for example, particular dates are defined, or iterative, where the context of each iteration will need to be considered.
  + Deciding what the documentation for the test project will be; for example, which plans, how the test cases will be documented and so on.
  + Defining the management information, including the metrics required, and putting in place the processes to monitor and control test preparation and execution, defect resolution and risk issues.
  + Ensuring that the test documentation generates repeatable test assets, for example, test cases.

1. **ENTRY CRITERIA AND EXIT CRITERIA (DEFINITION OF ‘READY’ OR DEFINITION OF ‘DONE’)**

* Entry criteria (also known as definition of ‘ready’ in Agile projects) define the preconditions for undertaking a test activity.
  + Availability of testable requirements, user stories or models.
  + Test environment available and ready for use (it functions).
  + Test tools installed in the environment are ready for use.
  + Testable code is available.
  + All test data is available and correct.
  + All previous test activity has been completed and met its exit criteria.
* Exit criteria are used to determine when a given test activity has been completed or when it should stop, typically called the definition of ‘done’ in an Agile project.
  + All tests planned have been executed.
  + A certain level of coverage has been achieved.
  + The number of unresolved defects is within an agreed limit.
  + All high-risk areas have been fully tested, with only minor residual risks left outstanding.
  + Cost – when the budget has been spent.
  + The number of estimated remaining defects is sufficiently low.
  + The evaluated level of quality criteria, such as reliability and performance, is sufficient.
  + The schedule has been achieved; for example, the release date has been reached and the product has to go live. This was the case with the millennium testing (it had to be completed before midnight on 31 December 1999) and is often the case with government legislation.

1. **TEST EXECUTION SCHEDULE**

* A test execution schedule documents what test suite will be run in what order and on what day. The order of the execution of test suites will be determined by many things such as prioritization, processing dependencies; for example, suite 1 has to run before suite 7 can be run, whether there are confirmation and regression tests, and finally in the most efficient sequence possible.

1. **FACTORS INFLUENCING THE TEST EFFORT**
2. Product Characteristics

* The risks associated with the product (defined during risk-based testing).
* The quality of the test basis; for example, requirements, user stories and so on.
* The complexity of the product domain.
* The number of quality characteristics such as reliability.
* The number of non-functional requirements.
* The security requirements (perhaps meeting ISO 27001, the security standard).
* How much documentation is required (e.g., some legislation-driven changes demand a certain level of documentation that may be more than an organization would normally produce).
* Requirements for legal or regulatory compliance.

1. Development Process Characteristics

* The stability and maturity of the organization; for example, a very process mature organization will take a lot less time to achieve what an immature (seat of their pants) organization would take, as they are likely to make less mistakes.
* The development model in use, such as Agile or sequential.
* The agreed test approaches.
* The tools in use, automation, test management and so on.
* The test process is defined in the test strategy and approach.
* Timescales.

1. People Characteristics

* The skills of those involved in the testing and development activity (the lower the skill level in development, the more defects could be introduced, and the lower the skill level in testing, the more detailed the test documentation needs to be).
* Team cohesion and leadership.

1. Test Results

* The number and severity of defects expected to be found.
* The amount of rework needed.

1. **Test Estimation**
   1. **The Metrics-Based Approach**

* This approach relies on data collected from previous or similar projects. This kind of data might include.
  + the number of tests conditions.
  + the number of test cases written.
  + the number of test cases executed.
  + the time taken to develop test cases.
  + the time taken to run test cases.
  + the number of defects found.
  + the number of environmental outages and how long on average each one lasted.
* With this approach and the right data, it is possible to estimate quite accurately what the cost and time required for a similar project would be.
* It is important that the actual costs and time for testing are accurately recorded. These can then be used to revalidate and possibly update the metrics for use on the next similar project.
  1. **The Expert-Based Approach**
* This alternative approach to metrics is to use 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:
  + business experts
  + test process consultants
  + developers
  + technical architects
  + analysts and designers
  + anyone with knowledge of the application to be tested or the tasks involved in the process.

1. **TEST MONITORING AND CONTROL**

* The purpose of test monitoring is to provide feedback and visibility of the progress of test activities.

1. **Metrics Used in Testing**

* In any project, metrics can be collected at any time – either during or at the end of the project and includes percentage of planned work in test case preparation, test environment preparation, test case execution, detect information, test coverage of requirements, subjective confidence of tests in the product, task completion, dates of test milestones, and testing costs.

1. **Test Reporting**

* Test reporting is the process whereby test metrics are reported in a summarized format both during and at the end of a test activity, to update the reader regarding the testing tasks undertaken.
* Test reports produced during the test activity are referred to as test progress reports, whereas a test report produced after a test activity has completed may be referred to as a test summary report.

1. **DEFECT MANAGEMENT**

* In testing, this translates into anything where the actual result is different from the expected result.
* Defect management is the process of recognizing, investigating, acting, and disposing of defects.
* It involves recording defects, classifying them, and identifying the impact.
* The process of defect management ensures that defects are tracked from recognition to correction, and finally through retest and closure.
* It is important that organizations document their defect management process and ensure that they have appointed someone (often called a defect manager/coordinator) to manage/police the process.

1. **CONFIGURATION MANAGEMENT**

* For testing, configuration management will involve controlling both the versions of code to be tested and the documents used during the development process, for example, requirements, design, and plans.
* Effective configuration management is important for the test process as the contents of each release of software into a test environment must be understood and at the correct version, otherwise testers could end up wasting time because either they are testing an invalid release of the software or the release does not integrate successfully, leading to the failure of many tests.
* A good configuration management system will ensure that the testers can identify exactly what code they are testing as well as have control over the test documentation such as test plans, test specification, defect logs and so on.

1. **SUMMARY**

* In this chapter we have looked at the component parts of test management.
* We initially explored risk and testing.
* When developing the test plan, the test manager and tester will look at the product risks (risks that relate directly to the failure of the product in the live environment) to decide what is important to test, as well as ensuring that any project risks (risks relating to the delivery of the project) are mitigated.
* The importance of independence in the test organization and how independence helps to ensure that the right focus is given to the test activity was reviewed.
* Independence is gained by separating the creative development activity from the test activity and we looked at the different levels of independence that are achievable:
  + the developers – low independence
  + independent testers ceded to the development team.
  + independent permanent test team, a center of excellence within the organization
  + independent testers or test team provided by the operational business unit.
  + outsourced test team or the use of independent contractors – high independence.
* We have looked at the test strategy and approach and how they shape the test activity based on many influences, including risks and the objectives of the testing.
* We have reviewed two roles that exist within a test project: test manager and tester. Both roles are important to the delivery of testing but could be invested in one or many people; for example, one person could have the role of test manager and tester.
* A test manager has responsibility for all of the planning activity, while the tester has responsibility for activities that surround the preparation of test cases. ISO 29119-3 provides outlines of four test-planning documents.
  + the test plans.
  + the test progress report
  + the test summary report
  + the test defect report.
* Test management depends not only on the preparation of the required documents but also on the development of the right entry and exit criteria and estimates, the monitoring of progress through the plan and the control activities implemented to ensure the plan is achieved.
* Test estimating can be achieved in one of two ways: metrics or an expert-based approach.
* After a plan of activity has been developed and time begins to pass, the test manager needs to monitor the progress of the activities.
* If any activity is delayed or there has been a change of any kind in the project itself, the test manager may need to revise the plan or take other actions to ensure that the project is delivered on time.
* We explored how the defects found during testing are recorded, and we reviewed the level of detail that needs to be recorded to ensure that any defect is fully understood and that any fix then made is the right one.
* Finally, we looked at configuration management. When running test cases against the code, it is important that the tester is aware of the version of code being tested and the version of the test being run. Controlling the versioning of the software and test assets is called configuration management.
* Lack of configuration management may lead to issues like loss of already-delivered functionality, reappearance of previously corrected errors and no understanding of which version of the test was run against which version of the code.