# Software Test Planning

# 5

**Key Topics**

Estimation
Work breakdown structure
Scheduling
Risk management
Project governance
Test reporting
Test monitoring and control

## 5.1 Introduction

Testing is a sub-project of a project and needs to be managed as such, and so good planning and monitoring and control are required. Test planning involves defining the scope of the testing to be performed; defining the test environment; estimating the effort required to define the test cases and to perform the testing; identifying the resources needed (including people, hardware, software, and tools); assigning the resources to the tasks; defining the schedule; and identifying any risks to the testing and managing them.

Test monitoring and control involve monitoring progress and taking corrective action when progress deviates from expectations; re-planning where the scope of the testing has changed; communicating progress to the various stakeholders with test reports to provide visibility into the testing carried out; taking corrective action to ensure quality and schedule are achieved; managing risks and issues; managing the change requests that arise during the project; and providing a final test report



G. O'Regan, *Concise Guide to Software Testing*,
Undergraduate Topics in Computer Science,
https://doi.org/10.1007/978-3-030-28494-7_5

with a recommendation to go to acceptance testing. The effective management of testing involves:

– Defining the scope of the testing
– Determining types of testing to be performed
– Estimates of time, effort, cost, resources, people, hardware, software, and tools
– Determining the start and end dates for the testing
– Determining the resources and staffing required
– Determining how test progress will be communicated
– Defining how test defects will be logged and reported
– Definition of test environment
– Assigning resources to the various tasks and activities
– Preparing the test plan
– Scheduling the various tasks and activities
– Preparing the initial test schedule and key milestones
– Identifying the key risks to testing
– Monitoring progress, budget, schedule, effort, risks, issues, change requests, and quality and taking corrective action
– Re-planning and rescheduling
– Providing regular status of passed, blocked, failed tests
– Communicating progress to affected stakeholders
– Preparing status reports and presentations
– Re-planning if scope of the project changes
– Conducting post-mortem to learn any lessons from the testing.

The test plan for the project is documented (this could be part of the project plan, but it is often in a separate document based on the test planning template in the IEEE 829 standard). It includes the scope of the testing, the personnel involved, the resources and effort required, the key milestones, the definition of the test environment, any special hardware and test tools required, and the planned test schedule. There is a separate test specification plan for the various types of testing, which records the test cases, including the purpose of each test case, the inputs and expected outputs, and the test procedure for the execution of the particular test case.

Several types of testing are performed during the project, including unit, integration, system, regression, performance, and user acceptance testing. The software developers perform the unit testing to verify the correctness of a module. This type of testing is termed "*white box*" testing and is based on knowledge of the internals of the software module. It involves defining and executing test cases to ensure code and branch coverage. The objective of "*black box*" testing is to verify the functionality of a module (or feature or the complete system itself), and knowledge of the internals of the software module is not required.

Test reporting is an important part of the project, and it ensures that all project participants understand the current quality of the software, as well as understanding what needs to be done to ensure that the product achieves the desired quality

criteria. The test status is reported regularly during the project, and once the tester discovers a defect, a problem report is opened, and the problem is analysed and corrected by the software developers. The problem may indicate a genuine defect, a misunderstanding by the tester, or a request for an enhancement.

Table 3.3 presents a simple test schedule for a small project, and Microsoft Project is generally employed for the planning and tracking of larger projects (Fig. 5.2). The activities in the test schedule are tracked and progress updated to record the tasks that have been completed, with new dates applied to tasks that have fallen behind schedule. Testing is a key sub-project of the main project, and the project manager will track the key test milestones and will maintain close contact with the test manager.

The effective management of risk during testing is essential to project success. It is prudent to consider risk management early in test planning, to identify risks that could potentially arise during the testing, and to identify (as far as is practical) actions to mitigate the risk or a contingency plan to address the risk if it materializes.

Risks arise due to uncertainty, and the risk management cycle involves[1] risk identification; risk analysis and evaluation; identifying responses to risks; selecting and planning a response to the risk; and risk monitoring. Once the risks have been identified, they are logged (e.g. in the risk log). The likelihood of each risk arising and its impact should it materialize is then determined. The risk is assigned an owner and an appropriate response to the risk determined.

Estimation is difficult as software projects are often breaking new ground and differ from previous projects. That is, historical estimates may often not be a good basis for estimation for the current project. Often, unanticipated problems may arise for technically advanced projects, and the estimates may be overly optimistic.

*Gantt charts* are generally employed for project scheduling, and these show the work breakdown for the project as well as task dependencies and allocation of staff to the various tasks.

Two popular project management methodologies are the *PRINCE*2 methodology (Office of Government Commerce 2004), which was developed in the UK, and *Project Management Professional (*PMP*)* and its associated project management body of knowledge (PMBOK) from the *Project Management Institute* (PMI) in the USA.

The test manager works closely with the project manager during the project, with the project manager responsible for the day-to-day management of the project and the test manager responsible for the day-to-day management of the testing. The *project board* (or steering group) includes the key stakeholders and is accountable for the success of the project. The project manager provides regular status reports to the project board during the project, and the test manager liaises with the project manager to ensure that the key test status is presented. The project board is consulted when key project decisions need to be made.

[1]These are the risk management activities in the PRINCE2 methodology.

## 5.2 Test Estimation

Estimation is a key part of project planning, and the accurate estimates of effort, cost, and schedule are essential to delivering a project on time and on budget, and with the right quality.[2] Estimation is employed in the planning process to determine the resources and effort required, and it feeds into the scheduling of the testing. The problems with over- or underestimation of projects are well known, and good estimates allow:

– Accurate calculation of the cost of testing
– Accurate scheduling of the testing
– Measurement of progress and costs against the estimates
– Determining the resources required for the testing.

Poor estimation leads to:

– Testing being over- or underestimated
– Testing being over- or under-resourced (impacting staff morale)
– Negative impression of the test manager.

Consequently, estimation needs to be rigorous, and there are several well-known techniques available (e.g. work breakdown structures, function points, and so on). Estimation applies to both the early and the later parts of the project, with the later phases of the project refining the initial estimates, as a more detailed understanding of the testing is then available. The new estimates are used to reschedule and to predict the eventual effort, delivery date, and cost of the project. The following are guidelines for estimation:

– Sufficient time needs to be allowed to do estimation.
– Historical data is often employed.
– Brainstorming is often employed.
– The initial estimates are high level.
– The estimates should be conservative rather than optimistic.
– Estimates will usually include contingency.
– Estimates should be reviewed to ensure their adequacy.
– Estimates from independent experts may be useful.
– It may be useful to prepare estimates using several methods and to compare.

Project metrics for testing (Figs. 9.4 and 9.5) may be employed to measure the accuracy of the estimates for the test planning. These include:

[2]The consequences of underestimating a project include the project being delivered late, with the project team working late nights and weekends to recover the schedule, quality being compromised with steps in the process omitted, and so on.

– Effort estimation accuracy
– Budget estimation accuracy
– Schedule estimation accuracy.

Next, we discuss several estimation techniques including the work breakdown structure, the analogy method, and the Delphi method.

### 5.2.1 Estimation Techniques

Estimates need to be produced consistently, and it would be inappropriate to have an estimation procedure such as *"Go ask Fred"*[3], as this clearly relies on an individual and is not a repeatable process. The estimates may be based on a work breakdown structure, function points, or another appropriate methodology. There are several approaches to estimation (Table 5.1) including.

### 5.2.2 Work Breakdown Structure

This is a popular approach to estimation (*it is also known as decomposition*) and involves the following:

– Identify the deliverables to be produced during the testing.
– Estimate the size of each deliverable (in pages or #test cases).
– Estimate the effort (number of days) required to complete the deliverable based on its complexity and size, and experience of team.
– Estimate the cost of the completed deliverable.
– The estimate for the testing is the sum of the individual estimates.

The approach often uses productivity data that is available from previously completed projects. The effort required for a complex deliverable is higher than that of a simple deliverable (where both are of the same size). The test planning section of the project plan (or a separate test plan) will detail the deliverables/tasks to be carried out in each phase. It may include a table similar to Table 5.2.

## 5.3 Test Planning and Scheduling

A well-managed project has an increased chance of success, and good planning is an essential part of project management. There is the well-known adage that states *"Fail to plan, plan to fail"*[4]. The test manager and the relevant stakeholders will

[3]Unless "Go ask Fred" is the name of the estimation methodology or the estimation tool employed.
[4]This quotation is adapted from Benjamin Franklin (an inventor and signatory to the American declaration of independence).

**Table 5.1** Estimation techniques

| Technique | Description |
|---|---|
| Work breakdown structure | Identify the deliverables to be produced during the testing. Estimate the size of each deliverable (in pages or #test cases). Estimate the effort (number of days) required to complete the deliverable based on its size and complexity. Estimate the cost of the completed deliverable |
| Analogy method | This involves comparing the proposed testing with a previously completed project (that is similar to the proposed project). The historical data and metrics for schedule, effort, and budget estimation accuracy are considered, as well as similarities and differences between the projects to provide effort, schedule, and budget estimates |
| Expert judgment | This involves consultation with experienced personnel to derive the estimate. The expert(s) can factor in differences between past projects, knowledge of existing systems as well as the specific requirements of the testing |
| Delphi method | The *Delphi method* is a consensus method used to produce accurate schedules and estimates. It was developed by the RAND Corporation and improved by Barry Boehm and others. It provides extra confidence in the estimates by using experts independent of the test manager |
| Cost predictor models | These include various cost prediction models such as *COCOMO* and SLIM. The Costar tool supports COCOMO, and the Qsm tool supports SLIM |
| Function points | *Function points* were developed by Allan Albrecht at IBM in the late 1970s and involve analysing each functional requirement and assigning a number of function points based on its size and complexity. This total number of function points is a measure of the estimate for the testing |

consider the appropriate approach for the testing and determine whether the testing should be outsourced to a third-party supplier or whether the test group has the competence and resources to perform the testing internally. A simple process map for test planning is presented in Fig. 5.1.

The effort estimates are used in scheduling of the tasks and activities using a project-scheduling tool such as *Microsoft Project* (Fig. 5.2). The schedule will include the key test milestones, the activities and tasks to be performed as well as their associated timescales, and the resources required to carry out each task.

The test manager will create the project test plan and schedule, and track the schedule to completion. The test manager will update the project schedule regularly during the project. The project test plan defines how the testing will be carried out, and it generally includes sections such as:

– Scope of testing
– Types of testing to be performed
– Roles and responsibilities
– Key stakeholders
– Resources required (hardware and human)
– Training, knowledge, and skills required

**Table 5.2** Example of work breakdown structure for test estimation

| Lifecycle phase | Project deliverable or task description | Est. size | Est. effort | Est. cost |
|---|---|---|---|---|
| Requirements/design | Inspections | | 2 days | $1000 |
| Coding phase | Code inspections | 50 | 2 days | $1000 |
| | Unit test cases | | 2 days | $1000 |
| | Unit testing | | 2 days | $1000 |
| Testing | Prepare test plan | 20 | 3 days | $1500 |
| | Define test environment | | 1 day | $500 |
| | Set up test environment | | 2 days | $1000 |
| | System test specs | | 2 days | $1000 |
| | Inspection | | 0.5 day | $250 |
| | System testing | | 1 day | $500 |
| | Performance test specs | | 2 days | $1000 |
| | Inspection | | 0.5 day | $250 |
| | Performance testing | | 1 day | $500 |
| | Regression tests | | 2 days | $1000 |
| | Regression testing | | 1 day | $500 |
| | UAT specs | | 2 days | $1000 |
| | Inspection | | 0.5 day | $250 |
| | UAT | | 1 day | $500 |
| | Test reporting | | 2 days | $1000 |
| Total | | | 29.5 | $14750 |
| Contingency | 10% | | | $1475 |
| Total | | | | $16,225 |

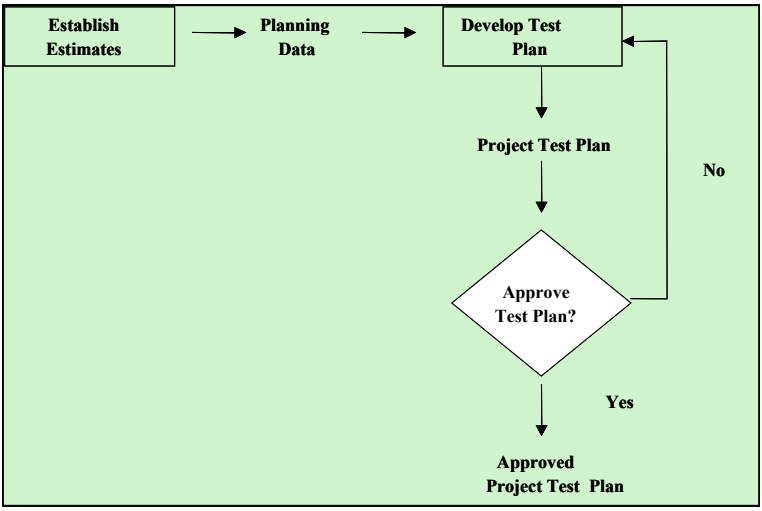
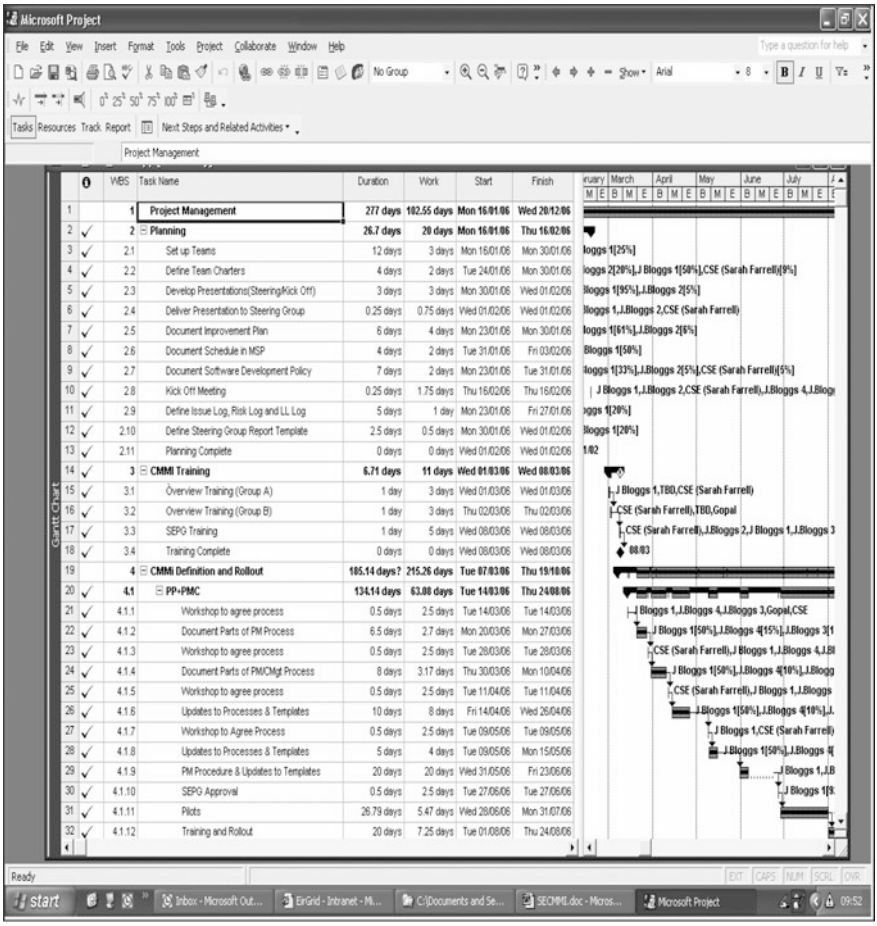



**Fig. 5.1** Simple process map for test planning




| | WBS | Task Name | Duration | Work | Start | Finish |
|---|---|---|---|---|---|---|
| 1 | 1 | **Project Management** | **277 days** | **102.55 days** | **Mon 16/01/06** | **Wed 20/12/06** |
| 2 | 2 | **Planning** | **26.7 days** | **20 days** | **Mon 16/01/06** | **Thu 16/02/06** |
| 3 | 2.1 | Set up Teams | 12 days | 3 days | Mon 16/01/06 | Mon 30/01/06 |
| 4 | 2.2 | Define Team Charters | 4 days | 2 days | Tue 24/01/06 | Mon 30/01/06 |
| 5 | 2.3 | Develop Presentations(Steering/Kick Off) | 3 days | 3 days | Mon 30/01/06 | Wed 01/02/06 |
| 6 | 2.4 | Deliver Presentation to Steering Group | 0.25 days | 0.75 days | Wed 01/02/06 | Wed 01/02/06 |
| 7 | 2.5 | Document Improvement Plan | 6 days | 4 days | Mon 23/01/06 | Mon 30/01/06 |
| 8 | 2.6 | Document Schedule in MSP | 4 days | 2 days | Tue 31/01/06 | Fri 03/02/06 |
| 9 | 2.7 | Document Software Development Policy | 7 days | 2 days | Mon 23/01/06 | Tue 31/01/06 |
| 10 | 2.8 | Kick Off Meeting | 0.25 days | 1.75 days | Thu 16/02/06 | Thu 16/02/06 |
| 11 | 2.9 | Define Issue Log, Risk Log and LL Log | 5 days | 1 day | Mon 23/01/06 | Fri 27/01/06 |
| 12 | 2.10 | Define Steering Group Report Template | 2.5 days | 0.5 days | Mon 30/01/06 | Wed 01/02/06 |
| 13 | 2.11 | Planning Complete | 0 days | 0 days | Wed 01/02/06 | Wed 01/02/06 |
| 14 | 3 | **CMMI Training** | **6.71 days** | **11 days** | **Wed 01/03/06** | **Wed 08/03/06** |
| 15 | 3.1 | Overview Training (Group A) | 1 day | 3 days | Wed 01/03/06 | Wed 01/03/06 |
| 16 | 3.2 | Overview Training (Group B) | 1 day | 3 days | Thu 02/03/06 | Thu 02/03/06 |
| 17 | 3.3 | SEPG Training | 1 day | 5 days | Wed 08/03/06 | Wed 08/03/06 |
| 18 | 3.4 | Training Complete | 0 days | 0 days | Wed 08/03/06 | Wed 08/03/06 |
| 19 | 4 | **CMMI Definition and Rollout** | **185.14 days?** | **215.26 days** | **Tue 07/03/06** | **Thu 19/10/06** |
| 20 | 4.1 | **PP+PMC** | **134.14 days** | **63.88 days** | **Tue 14/03/06** | **Thu 24/08/06** |
| 21 | 4.1.1 | Workshop to agree process | 0.5 days | 2.5 days | Tue 14/03/06 | Tue 14/03/06 |
| 22 | 4.1.2 | Document Parts of PM Process | 6.5 days | 2.7 days | Mon 20/03/06 | Mon 27/03/06 |
| 23 | 4.1.3 | Workshop to agree process | 0.5 days | 2.5 days | Tue 28/03/06 | Tue 28/03/06 |
| 24 | 4.1.4 | Document Parts of PM/CMgt Process | 8 days | 3.17 days | Thu 30/03/06 | Mon 10/04/06 |
| 25 | 4.1.5 | Workshop to agree process | 0.5 days | 2.5 days | Tue 11/04/06 | Tue 11/04/06 |
| 26 | 4.1.6 | Updates to Processes & Templates | 10 days | 8 days | Fri 14/04/06 | Wed 26/04/06 |
| 27 | 4.1.7 | Workshop to Agree Process | 0.5 days | 2.5 days | Tue 09/05/06 | Tue 09/05/06 |
| 28 | 4.1.8 | Updates to Processes & Templates | 5 days | 4 days | Tue 09/05/06 | Mon 15/05/06 |
| 29 | 4.1.9 | PM Procedure & Updates to Templates | 20 days | 20 days | Wed 31/05/06 | Fri 23/06/06 |
| 30 | 4.1.10 | SEPG Approval | 0.5 days | 2.5 days | Tue 27/06/06 | Tue 27/06/06 |
| 31 | 4.1.11 | Pilots | 26.79 days | 5.47 days | Wed 28/06/06 | Mon 31/07/06 |
| 32 | 4.1.12 | Training and Rollout | 20 days | 7.25 days | Tue 01/08/06 | Thu 24/08/06 |

**Fig. 5.2** Sample Microsoft Project schedule

– Key milestones (for testing)
– Schedule (for test activities, deliverables, and estimates)
– Key assumptions
– Key risks
– Communication planning and test reporting
– Budget planning
– Defect logging and retesting
– Test acceptance criteria
– Configuration management.

**Table 5.3** Sample test planning checklist

| No. | Item to check |
|---|---|
| 1. | Is the test plan complete and approved by the stakeholders? |
| 2. | Are estimates available for testing? Are they realistic? |
| 3. | Has the change control mechanism been set up for the project? |
| 4. | Are the risk log, issue log, and lessons learned log set up? |
| 5. | Are the responses to the risks and issues appropriate? |
| 6. | Have project and test communication been appropriately planned? |
| 7. | Are the key milestones defined? |
| 8. | Is the test schedule available? |
| 9. | Is the test schedule up to date? |
| 10. | Is the testing appropriately resourced? |
| 11. | Are all deliverables under configuration management control? |

There will be dedicated test plans for unit testing, system testing, and UAT. These are generally prepared as part of test case analysis and design which is described in Chap. 6, and we describe them briefly in Sect. 5.5.

Communication planning describes how communication will be carried out during the testing, and it includes the various meetings and reports that will be produced; financial planning is concerned with budget planning for the project (including the testing); configuration management is concerned with identifying the configuration items (i.e. the test deliverables) to be controlled and systematically controlling changes to them throughout the lifecycle (see Chap. 15). It ensures that all of the project deliverables are kept consistent following approved changes during the project.

The project test plan is a key project document, and it needs to be approved by the stakeholders. The test manager needs to ensure that the project test plan, the test schedule, and technical work products are kept consistent with the requirements. In other words, if there are changes to the requirements, then the project test plan and schedule will need to be updated accordingly.

Checklists are useful in verifying that the tasks have been completed. The sample checklist below (Table 5.3) may be tailored to verify that the test planning has been appropriately performed.

## 5.4 Risk Management in Testing

Risks arises due to uncertainty, and *risk management is concerned with managing uncertainty* and especially the management of any undesired events. Risks need to be identified, analysed, and controlled in order for the project to be successful, and risk management activities take place throughout the project lifecycle. There are risks that are specific to testing, and while the project manager has overall

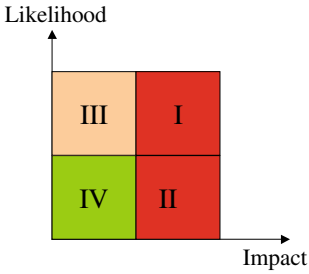



**Fig. 5.3** Risk categories

responsibility for risk management, in practice the test manager and project manager will work very closely on these risks.

Once the initial set of risks to the project has been identified, they are analysed to determine their *likelihood of occurrence* and their *impact* (e.g. on cost, schedule, or quality) should they materialize. These two parameters determine the *risk category*, and the most serious risk category refers to a risk with a high probability of occurring and a high impact on occurrence (i.e. Box I in Fig. 5.3).

Countermeasures are defined to reduce the likelihood of occurrence and impact of the risks, and contingency plans are prepared to deal with the situation of the risk actually materializing. Additional risks may arise during the testing, and the project manager/test manager need to be proactive in their identification and management.

Risks need to be reviewed regularly, especially following changes to the project. These could be changes to the business case or the business requirements, loss of key personnel, and so on. In other words, events that occur during the project may affect existing risks (including the probability of their occurrence and their impact) and may lead to new risks. Countermeasures need to be kept up to date during the project. Risks are reported regularly throughout the project.

Table 5.4 summarizes the activities in the risk management cycle including identifying risks; determining the probability of their occurrence and impact should they occur; identifying responses to the risks; and monitoring and reporting.

The project manager/test manager will maintain a risk repository (this may be a tool or a risk log) to record details of each risk, including its type and description; its likelihood and its impact (yielding the risk category); as well as the response to the risk. Sample risks to the testing in the project include:

– The software may be delivered late leading to the test schedule being cut short.
– The software may be of poor quality meaning that only limited testing may be done.
– A tester may resign.
– Specialized hardware required for testing may not arrive on time.
– The relationship between developers and testers may become antagonistic.
– The testers may lack the expertise to properly test the software.

**Table 5.4** Risk management activities

| Activity | Description |
|---|---|
| Risk management strategy | This defines how the risks will be identified, monitored, reviewed, and reported during the project, as well as the frequency of monitoring and reporting |
| Risk identification | This involves identifying the risks to the project and recording them in a risk repository (e.g. risk log). It continues throughout the project lifecycle. PRINCE2 classifies risks into:<br>– *Business* (e.g. collapse of subcontractors)<br>– *Legal and regulatory*<br>– *Organizational* (e.g. skilled resources/management)<br>– *Technical* (e.g. scope creep, architecture, design)<br>– *Environmental* (e.g. flooding or fires) |
| Evaluating the risks | This involves assessing the likelihood of occurrence of a particular risk and its impact (on cost, schedule, etc.) should it materialize. These two parameters result in the risk category |
| Identifying risk responses | The project manager/test manager will determine the appropriate response to a risk such as reducing the probability of its occurrence or its impact should it occur. These include:<br>– *Prevention* which aims to prevent it from occurring<br>– *Reduction* aims to reduce the probability of occurrence or impact should it occur<br>– *Transfer* aims to transfer the risk to a third party<br>– *Acceptance* is when nothing can be done about it<br>– *Contingency* is action that is carried out should the risk materialize |
| Risk monitoring and reporting | This involves monitoring existing risks to verify that the actions taken to manage the risks are effective, as well as identifying new risks. This helps in providing an early warning that an identified risk is going to materialize, and *a risk that materializes is a new project issue* that needs to be dealt with |
| Lessons learned | This is concerned with determining the effectiveness of risk management during the project and to learn any lessons for future projects |

## 5.5 Dedicated Test Plans

There will generally be specific test plans for the various types of testing performed such as unit, system, performance, and UAT. These plans specify how each type of testing will be performed, and they may include sections such as:

– Test objectives
– Approach
– Roles and responsibilities
– Key stakeholders
– Assumptions

– Risks
– Resources required
– Training required
– Preparation dates
– Testing dates
– Test environment
– Test tools
– Entry and exit criteria.

The dedicated test plan may contain a summary of the test cases to be executed as well as a traceability matrix that shows how the test cases cover the user or system requirements or design. These plans are written by the tester (or possibly the test leader depending on how the test team is organized). We discuss dedicated test plans again in Chap. 6.

## 5.6 Monitoring and Control

Test monitoring and control are concerned with monitoring test execution to give feedback and visibility on the test activities, and taking corrective action when performance deviates from expectations. The progress with the testing needs to be monitored against the plan, and corrective action taken when progress deviates from expectations. The key parameters such as effort and schedule as well as risks and issues are monitored, and the status of the testing is communicated regularly to the affected stakeholders.

The test manager will conduct regular progress and milestone reviews with the test team to determine actual progress and to identify new risks and issues. Figure 7.3 presents a simple process map for test monitoring and control, and the main focus is:

– Monitor the test plan and schedule.
– Monitor risks and issues and take appropriate action.
– Monitor resources and manage any resource issues.
– Conduct progress and milestone reviews.
– Measure test case execution, defect information, and test coverage.
– Re-plan as appropriate.
– Track corrective action to closure.
– Maintain close contact with the project manager and keep informed on progress.
– Prepare and present test reports detailing the test status.

The test manager is responsible for test monitoring and control, and for ensuring that appropriate corrective action is taken to address risks and issues. The status of the testing will be reported to the stakeholders in regular status reports.

### 5.6.1 Managing Issues, Change Requests, and Defects

The management of issues and change requests is a normal part of project management. An *issue* can arise at any time during the project (e.g. a supplier to the project may go out of business, an employee may resign, specialized hardware for testing may not arrive in time, and so on), and an issue refers to a problem that has occurred which may have a negative impact on the project. The severity of the issue is an indication of its impact on the project, and the project manager needs to manage it appropriately.

A *software defect* is a flaw in the software that causes it to produce an incorrect result, and it needs to be corrected by the developer and retested. The testers will identify defects during the various types of testing, and these are reported to the development team. The defect report should provide sufficient information to enable the developers to perform the necessary corrections, and the arrival rate of defects and the number of open defects provide an indication of the quality of the software. The management of defects is described in more detail in Sect. 7.4.

A *change request* is a stakeholder request for a change to the scope of the project, and it may arise at any time during the project. The impacts of the change request (e.g. technical, cost, and schedule impacts on development/testing) need to be carefully considered, as a change introduces new risks that may adversely affect cost, schedule, and quality. It is essential to understand these impacts to enable an informed decision on whether to authorize or reject the change request to be made. The project manager may directly approve small change requests, with the impacts of a larger change request considered by the project *change control board* (CCB).

The activities involved in managing issues and change requests are summarized below:

– Log issue or change request
– Assess impact
– Authorization (or rejection) of change request
– Implementation
– Verification
– Closure.

The management of change requests is discussed in more detail in Sect. 7.5.

## 5.7 Project Governance During Testing

The *project board*[5] (or steering group) is responsible for directing the project, and it is directly accountable for the success of the project. It consists of senior managers and staff in the organization who have the authority to make resources available, to remove roadblocks, and to get things done (Fig. 5.4).

It is consulted whenever key project decisions need to be made, and it plays a key role in project governance. The project board ensures that there is a clear business case for the project, and that the capital funding for the project is adequate and well spent. The project board may cancel the project at any stage during project execution should there cease to be a business case, or should project spending exceed tolerance and go out of control.[6]

The project manager reports to the project board and sends regular status reports to highlight progress made as well as the key project risks and issues. The project board meets at an appropriate frequency during the project (with extra sessions held should serious project issues arise).

The test manager will communicate the test status regularly to the project manager during the project, and the test status and the key test risks and issues will be discussed at the project board. The project manager attends the project board meeting and presents all key project information (including testing). There are several roles on the project board (an individual could perform more than one role), and their responsibilities are summarized in Table 5.5.

The project board will carefully consider the status of the project as well as the input from the project manager before deciding on the appropriate course of action (which could include the immediate termination of the project if there is no longer a business case for it).

## 5.8 Test Reporting

The frequency of test reporting is defined in the project test plan (or the communication plan). There is an IEEE standard (IEEE 829) for a test summary report. The test report advises management and the key stakeholders of the current status of the testing and includes key project testing information such as:

– Summary of testing activities and results
– Completed deliverables (during period)
– New risks and issues

[5]The project board in the PRINCE2 methodology includes roles such as the project executive, senior supplier, senior user, project assurance, and the project manager. These roles have distinct responsibilities.

[6]The project plan will usually specify a *tolerance level* for schedule and spending, where the project may spend (perhaps less than 10%) in excess of the allocated capital for the project before seeking authorization for further capital funding for the project.

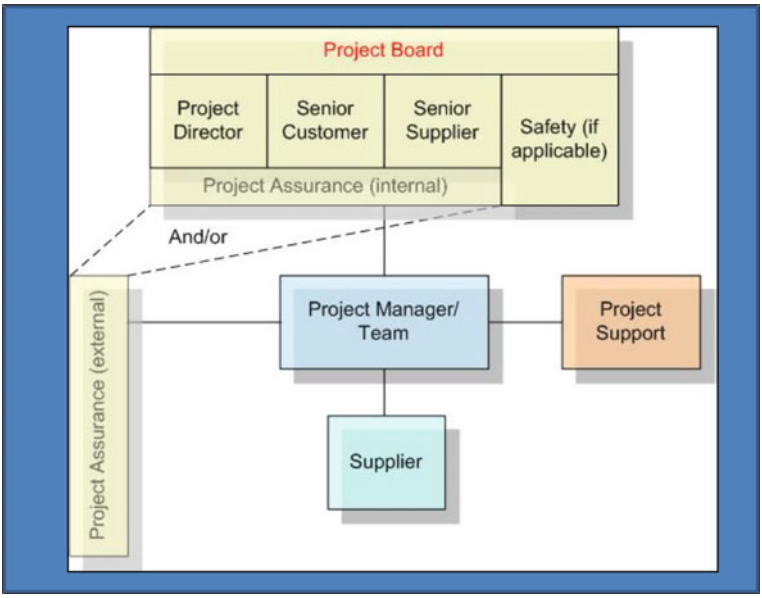



**Fig. 5.4** PRINCE2 project board

**Table 5.5** Project board roles and responsibilities

| Role | Responsibility |
|---|---|
| Project director | Ultimately responsible for the project. Provides overall guidance to the project |
| Senior customer | Represents the interests of users |
| Senior supplier | Represents the resources responsible for implementation of project (e.g. IS manager) |
| Project manager | Link between project board and project team |
| Project assurance | Internal role (optional) that provides an independent (of project manager) objective view of the project |
| Safety (optional) | Ensure adherence to health and safety standards |

– Schedule, effort, and budget status (e.g. RAG metrics[7])
– Test status
– Key risks and issues
– Milestone status
– Activities and deliverables planned (next period).

[7]Often, a colour coding mechanism is employed with a red flag indicating a serious issue; amber highlighting a potentially serious issue; and green indicating that everything is ok.

The test manager discusses the test report with management and presents the current status of the testing as well as the key risks and issues. The test manager will explain how the key issues are being dealt with and how the key risks will be managed. The new risks and issues will also be discussed, and management will carefully consider how the test manager plans to deal with these, and will provide appropriate support. The project manager will present a recovery plan (exception report) to deal with the situation where the project has fallen significantly outside the defined project tolerance (i.e. it is significantly behind schedule or over budget).

## 5.9 Lessons Learned and Project Closure

A project is a temporary activity, and once the project goals have been achieved and the product handed over to the customer and support group, it is ready to be closed. The project manager will prepare an end of project report detailing the extent to which the project achieved its objectives. The report will include a summary of key project metrics (including key quality metrics and the budget and timeliness metrics).

The success of the project is judged on the extent to which the defined objectives have been achieved and on the extent to which the project has delivered the agreed functionality on schedule, on budget, and with the right quality. This is often referred to as the project management triangle (Fig. 5.5).

The project manager presents the end project report to the project board, including any factors (e.g. change requests) that may have affected the timely delivery of the project or the allocated budget. The project is then officially closed.

The project manager and project team then consider the lessons learned during the projects, which are typically recorded in a lesson learned log. The key lessons learned are summarized in the lessons learned report, and the report is made available to other projects (with the goal of learning from experience). Any actions identified are assigned to individuals and followed through to closure. The project team is disbanded, and the project team members are assigned to other duties.

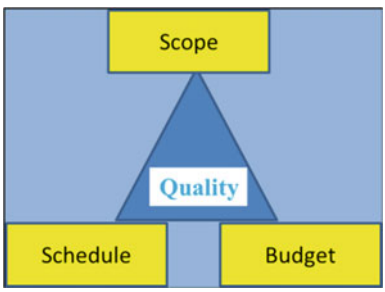



**Fig. 5.5** Project management triangle

## 5.10 Configuration Management

Configuration management is concerned with establishing and maintaining the integrity of the deliverables throughout the development lifecycle. It is concerned with:

– Configuration identification
– Configuration control
– Configuration control board (CCB)
– Baselining.

All test deliverables are uniquely identified and controlled. They are placed under configuration management control, including version control and change management. More detailed information on configuration management is in Chap. 15.

## 5.11 Review Questions

1. Describe the main activities in test planning?
2. Describe various approaches to estimation.
3. What skills are required to be a good test manager?
4. Explain the difference between the project test plan and the specific test plans for unit, system, and UAT.
5. What is the purpose of the project board?
6. What is the purpose of risk management? How are risks managed?
7. What is the difference between a risk and an issue?
8. How are defects handled during a project?
9. What is the purpose of test reporting?

## 5.12 Summary

Testing is a sub-project of a project and needs to be managed as such, and so good planning and monitoring and control are required. Test planning involves defining the scope of the testing to be performed; defining the test environment; estimating the effort required to define the test cases and to perform the testing; identifying the

resources needed; assigning the resources to the tasks; defining the schedule; and identifying any risks to the testing and managing them.

The project test plan is developed and approved by the stakeholders, and maintained during the project. Estimation and scheduling are difficult as software projects are often complex and quite different from previous projects. Gantt charts are often employed for scheduling, and these show the work breakdown for the project, as well as task dependencies and the assignment of staff to the various tasks.

The effective management of risk is essential to project success. Risks arise due to uncertainty, and the risk management cycle involves risk identification; risk analysis and evaluation; identifying responses to risks; selecting and planning a response to the risk; and risk monitoring.

Once the test planning is complete, the focus moves to monitoring progress, managing risks and issues, re-planning as appropriate, providing regular progress reports to the project board, and so on. Finally, there is an orderly close of the project.

## Reference

Office of Government Commerce (2004) Managing successful projects with PRINCE 2. The Stationary Office, London