1

Testing and Test Management

Automated teller machines that issue the wrong banknotes, Space Shuttles that crash on landing, telephone networks that go down and websites that are vulnerable to hackers are all familiar examples of ICT systems that fail, and there are new examples almost every day.

Testing is attracting more interest now than ever before, and test management has an important role to play in realising the considerable added value that testing can bring in controlling ICT quality and risks. Professional test management is an essential condition for this, and this book describes the practical approach and model developed by LogicaCMG for setting up professional test management. The model forms the thread throughout the book. Before presenting an overview, this chapter will place test management in the perspective of developments in the testing profession.

1.1 The Concept of Test Management

Over recent decades, the consequences of faulty ICT systems have greatly increased, as a significant proportion of our activities, both at work and in our private lives, have become dependent on ICT. After money, people and equipment, information is now seen as the fourth business resource (Mors, 1994).

It is not surprising, therefore, that the failure of ICT systems has come to have a large financial impact. In addition to the sensational examples in space exploration, there are plenty of others nearer to home. Faults in electronic service provision, such as placing an order or banking through the Internet, easily lead to lost revenue, expensive complaints handling and corrective measures, to say nothing of the damage to the reputation of the organisation.

It is therefore logical that organisations put a lot of effort into getting the quality of their information systems up to the required level. Ensure that

these systems will do what they have to do and not what they should not. The saying "prevention is better than cure" is particularly apt here. Testing is an information provider when delivering information systems because it is one way of measuring the quality of software. It is also a way of reducing risk of potential failures for an organisation. Some might even say that if there is no risk then there is no need to test. Testing consequently plays a crucial and essential role.

> Outside of ICT, too – among users and business managers, for example – the importance of testing is becoming recognised and is increasingly called upon.

Along with this, there has also been a natural increase in the importance of effective test project management. Many projects take too long and involve high costs, and even at the end of the project the information system still leaves something to be desired. On the other hand, intensive testing is carried out, even though it is unclear which risks are actually covered and it is difficult to say whether the information system actually will serve its intended purpose. It is all too common that a manager who is responsible for testing has no idea of how far he has progressed and is unable to indicate exactly how long his project will take to complete.

Bearing prime responsibility for testing, the test manager has a vital role to play in measuring quality. He does this by providing an insight into the risks involved in implementation and by supplying information at any point in time to senior management to facilitate informed decisions. Therefore, the test manager has to undertake close and regular monitoring. His sphere of influence goes beyond the scope of the test project itself.

The test manager directs the test project with a sharp focus on the interests of all the parties involved, starting with that of the client.

A professional level of test management means, among other things, the following:

- Test management has a clear and accepted position within the organisation and within improvement projects, including the corresponding responsibilities and competences.
- Test organisation complements the organisational set-up and takes account of all the available (central) testing facilities.
- The test manager communicates in a language that is understood by all parties involved. He provides management information to the client and other interested parties, clarifying what is being tested and to what extent. He consults with everyone concerned and indicates what the consequences of the choices made will be. System parts with high priority are tested as early as possible. Subsequently, with the aid of collected information, decisions can be made on the level of testing required for other parts, based on the defined minimum level of quality and required risk coverage.

- The test project is supported by a balanced budget and solid planning, enabling informed statements concerning the progress of the project.
- Test management facilitates the development and maintenance of metrics during the course of the project, in order to enhance the efficiency of testing (test process improvement).
- Targeted feedback from test management will help developers ensure that subsequent releases contain fewer design and program faults.

1.2 Evolution of Testing

It is useful to look at recent history and to consider the evolution of testing from that perspective. Testing came to maturity in many organisations under the influence of large projects, such as the millennium changeover and the preparations for introducing the euro. However, it is still only a short time ago that testing was left until the moment that system development was ready. There was no separate testing team, and testing was an activity that developers had to do in addition to their normal work. By several stages, testing has evolved into a profession in its own right. The following section describes some of the important phases of this development since the end of the 1970s.

1.2.1 "Intuitive" Testing

In the first phase, software developers are given plenty of time to write their programs, which they test themselves. When the system is ready, the project leaders invite a number of end users to test it over a few days, often at the weekend. If they report no critical issues and have a good feeling about it, the system goes into production on Monday morning. Far from testing being a separate phase in system development, it is almost a social event for a few end users. In the worst case, testing will show that the system completely fails to meet requirements. However, since it has been left to the end, there is no opportunity to make adjustments!

The question is, of course, what are the critical issues and what is the "good feeling" based upon? End users have their own expectations when they start the test. What one person considers extremely important may be of little interest to another. Their knowledge and experience may also vary widely. Do the end users involved have an overview of the full application, or has each individual only been involved in a part of it? Each one may give a different answer to the question of whether the quality of the application is good enough to go into production.

Testing, as described here, is still taking place in an unstructured manner. The project leader assumes that each end user knows what is important and will test accordingly. There is no test documentation, so at the end of the

test period it is not clear which actions were carried out. Perhaps various end users have tested the same system functions? Worse, whole sections of the information system may not have been covered at all during the tests. That is difficult to determine in retrospect: nobody documented their tests.

Resolving the issues can also present difficulties.

While entering a new order for a customer, an end user sees a button that will take him to the customer's previous order. After clicking on this button, an overview appears of all the orders arranged by date. The screen shows a menu option of an alphabetical arrangement. Interesting . . . so the end user performs the action. By clicking on an order, the invoice amount appears. The application can also flag up unpaid invoices. The end user has by now performed many actions, but only meant to enter a new order! When he tries to return to the order entry screen, something goes wrong and the application crashes.

In the absence of test documentation, replication of this issue is difficult. Does the end user remember the exact sequence of the steps he carried out? Perhaps he has forgotten, and is unable to replicate the issue. However, developers need this information to locate the problem. It is the only way they can get to a solution and improve the information system. Also, the degree of importance of an issue depends on the end user. If he works regularly with the tested functions, he will indicate that the issue is important, but he will find it of less importance if it is outside his field of operation. Nevertheless, these issues too might very well lead to big problems for the organisation as a whole.

Often, the clients decision is based on tests, while being unaware of how extensive they were, and the test findings provide insufficient insight into the quality of the system. Fortunately, this manner of testing is rare nowadays.

1.2.2 Testing with Design Documents

The next phase in the evolution of testing sees the use of design documents as a basis of testing. The testers begin by collecting documentation, usually functional and technical designs. Responsibility for the accuracy of the designs lies with the development team. The testers judge the design documents mainly on their suitability as a basis for testing. Often, testers start at the beginning of the design documents. For each function the testers come across, they design the required tests and carry on until they have dealt with all the functions, or, more likely, until the available time runs out.

The design of the functionality of an information system usually follows a logical sequence, which provides no indication of the testing priority. Neither are the connections within a design always clear. This means that there is only a small chance that the most important aspects have been tested by the end of the test cycle.

There is also a positive side. The use of design documents makes it easier for the testers to create test documentation. It gives them more to go on than the "intuitive" testing method, and at the end of the test they are better able to indicate what they have tested. The issues are resolved easier by the developers, since the testers document the steps taken. The outstanding issues are easier to place, since it is known which tests brought them to light.

This method of testing therefore provides the test manager with more material on which to base his recommendations, in comparison with the "intuitive" testing method. Even so, this advice, too, contains a random element. For if the test cases have not all been carried out, who is to say that the most important ones have been? And how does one know that the issues still outstanding will not create the biggest problems?

1.2.3 Requirement-Based Testing

A subsequent improvement in the quality of testing is possible if requirements are taken as the basis for testing. Requirements are the most important specifications that are defined for the project, both for the business and for ICT. This provides the testers with better indications for preparing the tests than design documents. Requirements give more insight into the priorities of the client and other involved parties, and the testers can take particular account of these.

However, there is a hidden danger here, too. The testers follow the list of requirements. For each requirement, they search for the corresponding descriptions in the design documents and prepare the tests with these. At the end, test cases are set up for all the requirements and the test execution can start. Naturally, as always, time is limited. Starting test preparations from the top of the list of requirements can mean, for all kinds of reasons, that not all of the requirements have been worked out when the allotted time is up.

At the end of the test execution, the testers can indicate more accurately what the quality of the system is, as they know to what extent each individual requirement has been met. However, they do not know which requirements are wrong or have been forgotten – they are guided by the list of requirements, and a wrong interpretation of the requirements on their part leads to incorrect conclusions.

This type of requirement-based testing is still very much focused on the question: does the product meet the specifications?

Reviews or inspections can improve the quality of the requirements, but it is very difficult to make a complete list of requirements. Other measures are necessary to cover this problem. Risk & Requirement Based Testing is such a solution.

In many projects, the requirements are not fully available at the start of the test project. Such projects tend to be characterised by a multitude of

change requests: after all, the end users are discovering more and more what the application means to them. A professional change management procedure is therefore essential!

Torpedoes were developed for a submarine with a launch installation. The following requirement was defined: after launching, the torpedo had to seek a heat-emitting object. With this requirement in mind, the testers set to work. It was decided to test under safe conditions, i.e. in the middle of the sea and with an unmanned submarine. This turned out to be a good choice. Out at sea, the torpedo could not find a target other than the submarine that launched it, because the submarine radiated heat. The torpedo turned around and headed for the submarine. The requirement was met; the test was a success. The developers had built exactly to specifications, but it was hardly the desired result! This incorrectly formulated requirement was only revealed during the test.

In this specific case, the requirement concerned was modified after the first test: if the torpedo were to turn 180 degrees, it would explode. The testers quickly carried out this version, only to meet with disaster again: the torpedo jammed in the launcher. The submarine had to turn around to get to harbour...

Fortunately, the third version succeeded.

By making use of the requirements, the test results come closer to fulfilling the needs and wishes of the stakeholders. This enables the test manager to report better on the status of testing. He provides the test results per requirement, and can also indicate what has not been tested. The testers can link the issues with the requirements, and outstanding issues give the client an idea of the quality per requirement. Should a requirement have too many outstanding issues, then the stakeholders can decide to carry on with testing until every requirement has been met to an acceptable level of quality.

Testing is even better when the stakeholders assign priorities to the requirements. The testers can then focus on the requirements with the highest priority, and issues are given the same priority as the requirement that is related to the test case with which the issue was found. In this way, the developers can see which issues they need to resolve first. At the end of the test project, the client is better able to determine the quality of the application. He can at least assess whether the requirements with the highest priority have had all their issues resolved. For requirements with a low priority, he is able to see the nature of the outstanding issues. Based on this information, he can take a decision on either to go into production or to continue with testing.

1.3 The Challenge for Test Management

In the previous paragraphs, we looked at the phases in the evolution of testing in the recent past. This mainly focused on testing itself, rather than on controlling a test project. The first part features significantly in the literature on testing. Emphasis is on the different phases of the test project, on the techniques to be employed, while the management of the test project is dismissed with a few sentences. That is a missed opportunity.

In a transport company, a standard budget is allocated per project to testing, i.e. 30% of the development budget. This is based on past figures in the traditional COBOL environment. A project is now being started with the aid of Visual Basic. In this environment, development is faster, and less time is allowed for testing. However, the amount of functionality that has to be tested is the same as in the COBOL projects, meaning that a proportionately higher test budget is required. Since this was not planned for, the test manager runs into difficulties. The expectations do not match the means available to him.

A bank has decided to introduce automated testing. It is a good idea, since many regression tests will take place in future releases. The test manager takes proper account of this in his plans. However, owing to limited experience in this area, there is a delay in the development of automated test scripts, which holds up test execution. With testing on the critical path it eventually forms a bottleneck in the project. The implementation date is not achieved. How could the test manager have prevented this situation from occuring?

The test manager has the job of realising the test project on time and within budget. He wants to be able to show at any point how far testing has progressed as well as the level of quality the information system has reached. In this way, if the risk appears to rise above an acceptable level, he can take timely measures. The test manager is expected to be of considerable assistance to the client when it comes to deciding whether or not to put the system into production. By showing which risks have been covered and which ones are outstanding, he can assist the client to make better decisions. He provides an insight into the quality of the information system. The test team has tested the most important aspects at every point by starting with those that represent the highest risk.

Test management has its own assignment and its own responsibilities besides project management. This means that a test manager must be proactive and self-assured, and should not accept without question that the planning and budget for the test project are set by the project manager.

This is part of the remit of the test manager, though naturally in consultation with the project manager, so that the best possible conditions exist for the project.

The answer to this challenge for test management lies in the following two concepts:

1. Testing based on Risk & Reqirement Based Testing, whereby product risks are leading and linked to the requirements.
2. The professionalisation of test management within the organisation.

These concepts are an integral part of the test management approach of LogicaCMG. The model is the thread through the rest of this book. The following section describes the model and gives an overview of all the activities carried out by a test manager within a test project. The core is Risk & Reqirement Based Testing, which places risk management at the centre of testing. The next chapter expands on this. Test management can also be seen as the shell surrounding the test process. The set-up of the test project has no effect on the test management approach described. It is a general approach that can be used in combination with different test methods. This can be compared with the project management approach PRINCE2.[1] With this, it does not matter whether a system development project is phased with, for example, the systems development method (SDM) or the rational unified process (RUP). In that sense, comparison can be be made between PRINCE2 and the Test Management Model. This comparison is included in Appendix A.
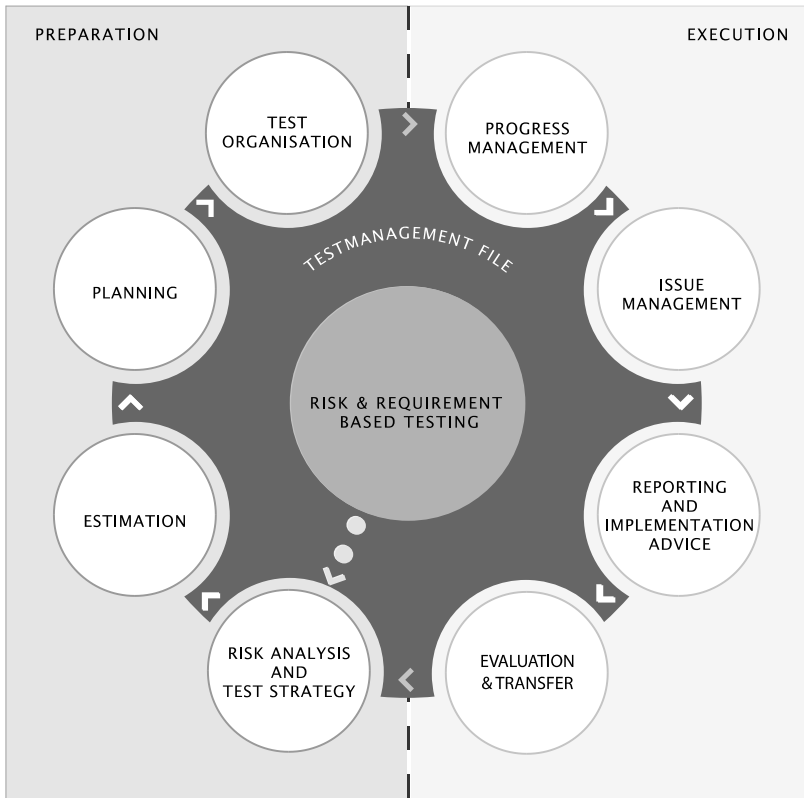
1.4 The Test Management Model

LogicaCMG's Test Management Model is characterised by the following points (see Figure 1.1):

- The model is an integral unit of coordinated activities. It is an approach that offers support from the beginning of a test project up to the moment of implementation and transfer.
- The model is made up of modular units. While the integration of the management activities delivers considerable added value, it is also possible to introduce a part of the activities into an organisation. Introduction of the approach can therefore be tailored to every organisation.
- The model is dynamic and iterative. This means that during the course of a project, it is possible to go back to earlier phases; for example, to adjust the test strategy in the event of altered product risks. This allows experience to be directly applied to the project.

---

[1] PRINCE2 is a worldwide standard for project management.

Fig. 1.1. The Test Management Model

- There is a division between the test project preparation activities of the test manager (i.e. planning, choice of strategy and the development of management instruments) and the test project execution activities of the test manager (the actual managing and communication of the status of the test project). Test management execution includes managing all the operational project activities, such as the development of test cases and automated test scripts, as well as managing test execution.
- There is a central role for risk management, and Risk & Requirement Based Testing is the approach to this. It is also the core of the Test Management Model. It is carried over to all the test management activities: the strategy is established based on the risks; progress management is based on the number of risks covered; reporting shows the outstanding risks and which requirements have already been covered.
- The model is aimed at the quick delivery of added value to the organisation. No lengthy preparations, but fast delivery of products that the organisation can immediately put to use. The most important product

that test management delivers is insight through clear management information, on the basis of which sound decisions can be made by the stakeholders concerning the implementation of an information system.

- The model accommodates a number of existing methods and best practice. Where this is applicable, this book provides references to the relevant literature. Examples are the references to PRINCE2, the earned value method and evolutionary planning.

To illustrate the model, and as an introduction to the following chapters, a brief description of each part of the model is given below.

1.4.1 Risk & Requirement Based Testing (RRBT)

The scope and depth of a test are determined by the risks that the organisation may run when putting a system into production. Working with a combination of product risks and requirements demonstrates where gaps lie.

The most important advantages of this approach are as follows:

- The focus lies on risks to the business instead of just the (described) functionality of the information system.
- The testing activities are prioritised based on product risks. The product risks with the highest priority can be tackled first.
- By focusing on product risks, overlooked or faulty requirements reveal themselves, so that omissions can be caught and dealt with at an early stage.
- During testing, communication takes place in a language (risks) that all stakeholders understand. The provision of high-value management information to the client and other stakeholders is a core task of the test manager. Most organisations already think in terms of (product) risks. Since reporting will be fit in with this, involvement in testing is greater than when a mere list of issues is communicated.

Risk & Requirement Based Testing (RRBT) is described in Chapter 2.

1.4.2 Test Management File

All documentation used and produced by the test manager and the test team is archived in the test management file. This is a directory structure to which all project members have access with different authorities. It is therefore possible at all times to check on which decisions were made based on what information. The test management file, which is described in Chapter 4, makes a test project easily transferable.

The left side of the model (**PREPARATION**) shows all the activities related to the preparation of the test project. During preparation, the

scope and depth of the tests are first determined, followed by estimating, planning and test organisation. This is in fact where the test project is defined.

1.4.3 Risk Analysis and Test Strategy

Within a test project, both the project risks and the product risks play an important role. Project risks are those that have a bearing on the execution of the project, and product risks are those that the organisation will face if the system does not function successfully. To define and manage these risks, the test manager has to carry out an analysis of both types. He incorporates the identified project risks together with their risk countermeasures into the test plan. The product risks are the basis of the test strategy. They are linked with the stakeholders of the project or the system. The test strategy also establishes which risks are covered by which tests. The test manager can use the strategy if the time available is less than planned. In this case, he can provide a documented indication of the effects that implementation of the system will have and which product risks cannot be assessed. He can also indicate which of the quantitative acceptance criteria contained in the strategy have been met. Based on that information, an informed decision can be made on whether to put the information system into production or to supplement the budget. Risk analysis and test strategy are detailed in Chapter 5.

1.4.4 Estimation

Many test managers wrestle with the problem of budget allocation for a test project. Since the scope of the test project is described in the test strategy, it can be used by the test manager as a basis for budget allocation. The allocation itself can take place in several ways. Chapter 6 describes various methods, with their advantages and disadvantages. This helps with finding a suitable approach to a certain situation, for each situation is unique. Chapter 6 introduces a pragmatic model, the test effort estimation model (TEEM), specially developed for test budgeting based on historical metrics (see Section 6.2.7).

1.4.5 Planning

The number of hours required for the test project is determined and laid down in the budget, allowing the test manager to plot the testing effort within the given time. He then establishes the necessary resources and plans first for the highest risk areas, so that if unforeseen circumstances lead to time pressure, the most important tests have been carried out.

A detailed plan of a complete project offers the illusion of accuracy, as the execution of a test project is subject to many influential factors. For that reason, evolutionary planning is the method of choice: the first phase of the project is surveyed in detail and a global plan is issued for the subsequent phases.

The planning, just as with PRINCE2, is product focused, so that added value is delivered to the organisation as quickly as possible. Instead of extensive preparation followed by execution of the tests, small units are developed including preparation and execution. This immediately demonstrates the status of particular parts of an information system. The planning for test projects is detailed in Chapter 7.

1.4.6 Test Organisation

In a test project, there is always a need for a good mix of technical, testing and business expertise to be available at the right time. Some organisations have central support for testing, such as a test competence centre or a test centre. The test manager has to take account of these facilities when organising the test.

The organisation of the test project is detailed in Chapter 8.

> The right side of the model (**EXECUTION**) shows all the test manager's activities that are concerned with the execution of the test project. This includes managing test preparations and test execution and all the surrounding factors that require attention, such as issue management, reporting, advice, evaluation and transfer.

1.4.7 Progress Management

Three "business" controls are available to the test manager for managing progress: time, money and quality. He needs to have continuous insight into the status of the test project: progress (time), budget exhaustion (money) and the outstanding and covered risks (quality). Chapter 10 discusses the earned value method (EVM), which is used to obtain and keep control of the test project. This method helps to monitor the business controls of time and money. With regard to reporting on quality, the results of the test execution are registered with the test cases. These test cases are related to the product risks that have to be covered. Managing based on the variables of time, money and quality is supported by the test control matrix (TCM), which is described in some detail in Chapter 10.

1.4.8 Issue Management

During test execution, the test team will encounter various issues. It is important that these are recorded accurately and that their status is actively monitored. The test manager requires this information for reporting on the quality of the information system. In setting up the test, a relationship is established between the test cases and the product risks. The priority of the product risks determines, via the test case, the priority of the issue. This gives the client insight into the risks that are still outstanding in the system because of (as yet) unsolved issues.

Well-organised issue management also provides the developers with a means of investigating where and under what circumstances the issues manifest themselves. By also recording in the issue administration the particular phase of system development in which the issues should have been found or prevented, a basis is laid for process improvement – not only of the testing process, but also of the development process. Issue management is detailed in Chapter 11.

1.4.9 Reporting and Advice

During execution of the test project, it is important that the test manager keeps the client and all other stakeholders informed. At the end of the test project, or during the transfer from one test level to the next, he should provide a report with advice on the quality of the information system. Considering that quality is also discussed in the periodic progress reports, his conclusions should not come as a surprise. Information on the covered and outstanding product risks forms the core of the report. This delivers management information on which the organisation can base its decisions. Of course, project risks are also part of the reports. These often explain deviations between planning and realisation. Reporting and advice are detailed in Chapter 12.

1.4.10 Evaluation and Transfer

Evaluation takes place during and at the end of the test project. Interim evaluations are important for improving aspects of the current test project. The final evaluation provides an indication of the extent to which the formulated test assignments have been met. The final evaluation also provides insight into the possibilities for improvement in future projects. Such improvements may involve the development process as well as the testing process. During the test project, metrics can be maintained based on the goal–question–metric (GQM) method for the purposes of process improvement (see Appendix F).

When the test project is completed, the test manager will hand over the products in the test management file to a maintenance organisation. Before doing so, he will update the documentation to reflect the status of

the project. Because changes in insight and requirements during the project have an impact on the test strategy, it is important for the next release of the information system that these details are incorporated into the test strategy. Not all products of the test project will be maintained in the maintenance organisation; those that are not to be maintained are archived.

Evaluation and transfer are detailed in Chapter 13.

1.5 Conclusion

The test management activities described in the model form the baseline of this book. With each of the topics, reference is made to the core of the Test Management Model: namely, risks and requirements. By using this model, the test manager is constantly aware of these important aspects of the tests and can take them into account. He has a model to hand that will assist him to take the right steps and make the right decisions.

The following chapter first expands on the core of the model, i.e. Risk & Requirement Based Testing.