A

PRINCE2 and Test Management

PRINCE2 is a project management method that is generally applicable to projects. The name PRINCE stands for PRojects IN Controlled Environments. The PRINCE2 method was developed by the Central Computer and Telecommunications Agency (CCTA) as a process-based method that anyone can use. The CCTA is part of the British government and sets out "best practice" work methods. The method is based on practical experience with various management methods and pays particular attention to changes in environmental factors that might influence the success of a project. PRINCE2 is the standard in many organisations in Europe for managing (ICT) projects.

The test management approach described in this book bears many similarities to the PRINCE2 approach. Test management can therefore function perfectly well in an ICT project driven by PRINCE2. Within the practice of project execution, the two methods can effectively reinforce each other.

A.1 Similarities Between PRINCE2 and Test Management

We describe briefly here how a number of main principles of PRINCE2 compare with the test management approach described in this book. We also discuss how test management fits into a PRINCE2 project.

The PRINCE2 principles are summarised below, followed by a description of the way in which test management corresponds with these.

1. *The instigation of the project is a specified and measurable business case. Evaluation of this business case in the course of the project can lead to suspension of the project.*
   In the test management approach, the test process is set up and controlled by measures based on a risk analysis and on the requirements.

Both of these factors take the business case as the starting point. Test management can therefore offer considerable added value to the evaluation of the business case during the project. The test processes are set up in such a way that they can generate relevant and targeted management information, on progress and quality of the end product, among other things. With the aid of this information, the client (the business case owner) can determine whether the business case of the project is still valid. With an objective and quantified insight into the status of the end product, an estimate can be made, for example, of how much more time and money it will cost to realise an acceptable level of the information system. If the product is delivered too late, or if it appears to be too expensive, this may be reason for the client to halt the project.

2. *The business representation in a project plays a vital role that is not without duties.*
   A fixed part of the test management approach is carrying out a stakeholder analysis at the start of the project. In the analysis, the interests of all the parties involved in the project are surveyed. Subsequently in the test strategy, this is explicitly translated into the tasks, responsibilities and authorities that the parties are to be assigned to in the project execution. The results of this participation are established in as concrete a form as possible. In this way, the input of the business into the testing is secured, just as it is in PRINCE2.
3. *Risk management is one of the core processes within PRINCE2.*
   Risk management also takes a central place in the test management philosophy: as with PRINCE2, a sound risk analysis is carried out at the start of a test project. The risk approach of test management forms the basis of the substantive management of the test project. And it mainly concerns the product risks: which risks are directly related to the information system.
   Besides technical risks, this also concerns business-related risks. With these, the product risk analysis forms the essential input for the test strategy. The test manager monitors these product risks through progress management. He must also monitor the project risks that relate to the project itself.
4. *The planning process is product oriented and uses a product breakdown structure, among other things.*
   The product-oriented planning of the test management approach aligns perfectly with this. Planning on the basis of products is also an important principle of test management, since it allows concrete results to be delivered at an early stage. Testing focuses first on those products that are the most important to the client as regards risk and added value. With these priorities, test management is able to create "benefit-based" reports. These provide the client with an objective picture of the quality of the products.

5. *Plans are iterative in nature.*
   The test management approach also develops plans in an iterative way. The specific approach that is used within test management is based on “evolutionary planning” (Gilb, 1988). With this, learning experiences during the test project are used towards the most realistic planning of time possible for the remainder of the course of the test project.
6. *Explicit separation of project initiation (in PRINCE2 the IP phase) and project management (in PRINCE2 the BF phase).*
   Test management makes a clear distinction between activities during the preparation of a test project and those during the execution of it. See the Test Management Model.

A.2 Position of a Test Project Within a PRINCE2 Project

PRINCE2 projects have their own structure, based on a client–vendor relationship. An important premise of this is that there is always a client who will specify the required project result, who will use the project result, and who will most probably pay for the project. This makes it clear to the test organisation where the responsibilities lie, who can be called to account on which points, and what the escalation paths are.

Test management in a PRINCE2 project will take the place of what in PRINCE2 terms is called “Team Management”. Within that team (or project) the test organisation can be set up in accordance with the structure in the test management approach. The exchange of information with the rest of the project then runs according to the PRINCE2 approach. The team manager (in this case the test manager) will then make concrete agreements with the client or programme manager concerning the results to be delivered.

A.3 More PRINCE2

In various chapters of this book where the phases of the Test Management Model are discussed, references are made here and there to PRINCE2.

If you wish to know more about PRINCE2, you will find information in books (CCTA, 2002) or on the official PRINCE2 website: www.ogc.gov.uk/prince.

B

Checklist of Product Risks

This checklist contains a large collection of questions and answers that can help the test manager in determining the product risks concerning the implementation of an information system.

The questions are divided into the following categories, based on the most usual groups of stakeholders (see Chapter 5, Section 5.2.1):

- end users;
- marketing;
- support department (such as a helpdesk);
- IT department;
- internal security.

An example is provided per product risk together with the quality attribute within ISO 9126 that best applies to the relevant risk.

The last column is meant to indicate the priority per product risk under the MoSCoW rules (i.e. Must test, Should test, Could test or Won't test).

For example:

No.	**Product risk indication**	**Example**	**Quality attribute (ISO9126)**	**Importance (MoSCoW)**
1.2	Should the correlation between input fields in the information system be checked?	Are we checking whether the prescribed correlations between input fields are correct and complete? (Semantic)	Suitability	

B.1 Checklist of Product Risks – End Users

No.	Product risk indication	Example	Quality attribute (ISO9126)	Importance (MoSCoW)
1.1	Should the input be checked by the information system?	Should it be checked whether all the input fields comply with the prescribed "type" (numeric, alphanumeric, date), length and style (bold, underlined, italics, etc.)? Can correct values be entered, and are incorrect values rejected? (Syntax)	Accuracy	
1.2	Should the correlation between input fields in the information system be checked?	Should it be checked whether the prescribed correlation between input fields is correct and complete? (Semantic)	Suitability	
1.3	Are the users experienced with the user interface of the information system?	How easily can the end users work with a new version of the information system?	Operability	

1.4	Are the error messages and other messages displayed by the information system clear?	Do the error messages and other displayed messages provide the information needed by the users to understand which fault has occurred, or which action has been carried out and/or which action they themselves should take?	Understandability
1.5	Are the help facilities clear?	After starting up a help facility, can the end user operate it easily?	Learnability
1.6	Is the menu structure clear?	Is it clear to the end users how they can navigate through the information system and which functions they can perform with it?	Understandability
1.7	Are the screens clear and legible?	Do the end users quickly understand how and where they should carry out particular actions? Do the screens have a layout that appeals to the end users?	Attractiveness
1.8	Are the generated overviews clear?	Is the layout of the overviews clear to the end users?	Understandability

1.9	Do the overviews contain the required information?	Do the overviews provide the information that the end users require to carry out their tasks?	Suitability
1.10	Is the user manual clear?	Does the user manual provide the users with adequate and useful information?	Learnability
1.11	Are there links with other systems?	When a car insurance policy is concluded, is notification automatically sent to the government department for road transport?	Interoperability
1.12	Are there alternatives for processing data if the information system is not available?	For example, for call-back notes. If no alternatives are available, it must be possible to recover the system quickly	Recoverability
1.13	Should the system be available all day?	No down time should occur in a system that is critical to the company's operation 24 hours a day	Maturity
1.14	Should the information system be available in various languages?	If text in the information system has to be translated, mistakes may be created. Different languages can have different address forms, for example	Operability
1.15	Does peak load of the information system occur during the day?	At peak use, the response time of the information system may be longer	Time behaviour

1.16	Should the information system comply with a particular response time?	If a customer is waiting at the counter while the system is being used, it is important that the waiting time is not too long	Time behaviour
1.17	Do many users use the system simultaneously?	With simultaneous use of data, problems may arise concerning consistency of the database	Resource utilisation
1.18	Can the users adapt the system to suit their own preferences?	For example, the users themselves adapt the layout of reports or screens	Adaptability
1.19	Does the information system fit within existing standards and procedures?	Is the system suited to the daily tasks of the end users?	Suitability
1.20	Is the information system intended for processing financial details?	Financial details often require a high degree of accuracy	Accuracy
1.21	Are the details in the information system confidential?	With some information systems, the data must be used in confidence, e.g. bank account details	Security
1.22	Does the information system influence the physical environment?	For example, mission-critical systems, such as heart monitoring equipment	Maturity

1.23	Does the information system concern a primary or a secondary system for the business process?	With a secondary system, an occasional disruption is probably less serious than with a primary system	Maturity
1.24	Are the procedures for the business process stable?	If these often change, there is a risk that the information system is unsuitable for the changed process	Suitability
1.25	Does the information system affect other departments?	For example, a new quotation system may influence the printing channels	Co-existence
1.26	How many people use the system?	With many users, the information system will require more processing capacity	Resource utilisation
1.27	How many different users access the system?	These users will probably use the system in different ways. Have allowances been made for this in the user interface? For example, diverse menu structures and screens	Suitability
1.28	Does the information system have an online or a batch function?	With online systems, higher demands are made on the user interface	Operability
1.29	Is the entire business process covered by the information system?	Does a breakdown in the information system affect the whole of the company process?	Recoverability

1.30	Is the generation of management information the most important function of the information system?	Management uses the information for taking important decisions and will require a high degree of accuracy in this information	Accuracy
1.31	Are the users experienced in the use of information systems?	If the users previously did everything manually, more effort will be required in training and assisting them	Learnability
1.32	How much of the existing functionality from the previous version has been changed?	With many changes, the chances of errors are increased and the degree of end users acceptance may be reduced	Changeability
1.33	Should the information system store historical data?	Companies are obliged to retain transaction details for a certain number of years	Functionality compliance
1.34	Does the information system make complex calculations?	For example, calculation of premiums in an insurance quotation system	Accuracy

1.35	Can other information systems influence the operation of the system?	If another system goes down, can the target system continue operating? Are other systems given priority as regards processing if several systems are active simultaneously?	Co-existence
1.36	Is the processing of input time-critical?	Further processing could be done later if it is not time-critical	Time behaviour
1.37	Is the status of the information system always clear?	Does the information system indicate that it is busy processing? Is a message displayed when a transaction is completed or if a time-out occurs?	Understandability
1.38	Is the information system replacing an existing system?	The new system should be capable of supplying the existing functionality. Regression testing is important in that case	Replaceability
1.39	Should it be possible to operate the functions of the information system in various ways?	Varying the input instrument for repetitive functions can prevent repetitive strain injury (RSI)	Operability

B.2 Checklist of Product Risks – Marketing

No.	Product risk indication	Example	Quality attribute (ISO 9126)	Importance (MoSCoW)
2.1	Is the information system used for selling products?	For example, sales of insurance via the Internet	Operability	
2.2	Should the information system run on various platforms?	Internet applications can run on various operating systems, e.g. Netscape and Internet Explorer	Adaptability	
2.3	Does the information system comply with all the legal privacy regulations?	For example, Personal Data Protection Act	Functionality compliance	
2.4	Should clients be able to install the information system themselves?	When a system is sold to customers, it goes to a variety of end users with varying levels of knowledge and experience	Installability	
2.5	Is there a geographical distribution of the information system?	Is the system used only locally or, for example, worldwide?	Interoperability	

2.6	Is it important for (a new version of) the information system to be available for the market by a certain date?	If the time-to-market is short, the system should be easy to adapt	Changeability	

B.3 Checklist of Product Risks – Support Department

No.	Product risk indication	Example	Quality attribute (ISO9126)	Importance (MoSCoW)
3.1	Can the helpdesk provide status information?	Can the helpdesk supply insight into the status of the information system, and can it indicate how to proceed?	Analysability	
3.2	Is there a production manual?	Does the helpdesk have insight into the information system in production?	Operability	
3.3	Does the information system have an online or a batch function?	With a batch system, it is more difficult to trace errors	Analysability	
3.4	Do the batches have void lists?	Is it clear which transactions within a batch were unsuccessful?	Analysability	

B.4 Checklist of Product Risks – IT Department

No.	Product risk indication	Example	Quality attribute (ISO9126)	Importance (MoSCoW)
4.1	Is a summarised function description available?	This description aids the maintenance department with change implementation	Analysability	
4.2	Are back-ups created?	If the system goes down, can the backup be reinstalled?	Fault tolerance	
4.3	In the event of breakdown, should the information system be restored within a certain time?	Is there a maximum time during which the system is allowed to be unavailable?	Recoverability	
4.4	Is there a contingency plan if the hardware that runs the information system fails?	Is there a shadow system available that can take over the tasks?	Fault tolerance	
4.5	Has the information system been created with new technology?	Using technology that is new to the organisation can be the cause of a lot of errors	Maturity	

4.6	How do we deal with interrupted transactions?	Can the system revert to the situation as it was before the transaction was started?	Recoverability
4.7	Are there contingency plans for when the information system goes down?	Have we considered an emergency scenario?	Recoverability
4.8	Is the information system sensitive to settings on end users' PCs?	Can the users configure the system to their own preferences (e.g. resolution and screen size)?	Adaptability
4.9	Are data from other information systems stored in this system?	Are allowances made for synchronisation of the data between both systems?	Interoperability
4.10	Should the information system operate within the existing infrastructure?	The existing infrastructure may slow down when another system is added on. It is also possible that there is insufficient disk space available in the existing infrastructure	Co-existence

4.11	Are changes to the infrastructure planned?	If changes to the infrastructure will affect the system, they should be accommodated easily within the system	Adaptability
4.12	Are create, read, update and delete (CRUD) functions implemented for the various entities?	If these functions are not present for all the entities, some test cases cannot be carried out	Testability
4.13	Are correlations between entities checked upon the removal of an entity?	If a customer is displayed showing accounts attached, it should not be possible to delete the customer without administering these accounts	Accuracy
4.14	Are system components of third parties used?	Do these system components fit within the information system, or are there, for example, big differences in the layout of data?	Suitability
4.15	Is there a fallback scenario if the implementation of the information system fails?	Is an emergency plan available?	Fault tolerance

4.16	Is there a fixed time for processing batches?	Most companies have to process batches outside office hours. Users may not be logged in during batch processing. If the batch is not ready when an office opens, the employees cannot use the system	Time behaviour	
4.17	Do many errors occur in the information system when changes are introduced?	Changes can lead to unexpected errors, particularly with a legacy system where knowledge of the system is lacking	Stability	
4.18	Are there any set conditions concerning the time allowed to recover after a fault?	An organisation may set conditions concerning the maximum downtime a system is permitted. (Service level)	Recoverability	

B.5 Checklist of Product Risks – Internal Security

No.	**Product risk indication**	**Example**	**Quality attribute (ISOX9126)**	**Importance (MoSCoW)**
5.1	Are there guidelines for restricting access to the information system?	A procedure stating who may have what level of access to the system	Security	

5.2	Are passwords used?	Are there any rules concerning these passwords: length, change frequency, etc.?	Security
5.3	Are the access attempts monitored?	Is a log kept that registers who is trying to access the system and whether this person is authorised or not?	Security
5.4	Are internal development standards used?	Do the developers maintain these standards?	Maintainability compliance
5.5	Are there links to outside of the organisation?	Many links mean that there are more opportunities for hackers to enter the system	Security
5.6	Are transactions locked?	Locking the information makes it more difficult to use it for illegitimate purposes	Security
5.7	Is an audit trail required for the information system?	With many financial information systems, an audit trail is a legal requirement	Analysability

C

Template for Risk- & Requirement-Based Testing

The template below provides an overview of the information that is important within risk- and requirement-based testing (RRBT).

This table can be used in decision making.

If, for example, a requirement changes, the test manager can see to which product risk it is linked. He can also see the consequences it will have for the test condition(s). Are any adjustments to the test conditions required?

The table can also form the basis of the reports. See Chapter 12.

The testers can link an issue to the relevant product risk during the execution of the test via the test condition. With this, a tester can determine the initial priority of an issue.

The case from Chapter 2 has been used to give an idea of the way in which this table is completed.

No.	**Product risk**	**Quality attribute**	**MoSCoW**	**Stake-holder**	**Requirement**	**Test condition**
1	Customer cannot perform a transaction	Functionality	Must test	End user	Customer able to perform a transaction via own bank	It is possible to perform a transaction at own bank, using an existing pin code
						It is *not* possible to perform a transaction at own bank when using an invalid pin code
					Customer able to perform a transaction via other bank	
					Customer can choose from set amounts	
					Customer can select amount/choice of banknotes	
	…	…	…	…	…	…

D

Quality Attributes According to ISO9126

Quality attribute	Description
Functionality	**The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions**
Suitability	The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives
Accuracy	The capability of the software product to provide the right or agreed results or effects with the needed degree of precision
Interoperability	The capability of the software product to interact with one or more specified systems
Security	The capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access to them
Functionality compliance	The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality
Reliability	**The capability of the software product to maintain a specified level of performance when used under specified conditions**
Maturity	The capability of the software product to avoid failure as a result of faults in the software
Fault tolerance	The capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface
Recoverability	The capability of the software product to re-establish a specified level of performance and recover the data directly affected in the case of a failure
Reliability compliance	The capability of the software product to adhere to standards, conventions or regulations relating to reliability

Usability	**The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions**
Understandability	The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use
Learnability	The capability of the software product to enable the user to learn its application
Operability	The capability of the software product to enable the user to operate and control it
Attractiveness	The capability of the software product to be attractive to the user
Usability compliance	The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability
Efficiency	**The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions**
Time behaviour	The capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions
Resource utilisation	The capability of the software product to use appropriate amounts and types of resources when the software performs its function under stated conditions
Efficiency compliance	The capability of the software product to adhere to standards or conventions relating to efficiency

Maintainability	**The capability of the software product to be modified. Modifications may include corrections, improvements or adaption of the software to changes in environment, and in requirements and functional specifications**
Analysability	The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified
Changeability	The capability of the software product to enable a specified modification to be implemented
Stability	The capability of the software product to avoid unexpected effects from modifications of the software
Testability	The capability of the software product to enable modified software to be validated
Maintainability compliance	The capability of the software product to adhere to standards or conventions relating to maintainability
Portability	**The capability of the software product to be transferred from one environment to another**
Adaptability	The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered
Installability	The capability of the software product to be installed in a specified environment
Co-existence	The capability of the software product to co-exist with other independent software in a common environment sharing common resources
Replaceability	The capability of the software product to be used in place of another specified software product for the same purpose in the same environment
Portability compliance	The capability of the software product to adhere to standards or conventions relating to portability

Reference

ISO/IEC 9126-1, *Software engineering – Software product quality – Part 1: Quality model*, International Organization of Standardization, 2001.

E

Template for Test Plan

This appendix shows the subjects that should be contained within a test plan. This applies equally to a project test plan and a detailed test plan. General aspects such as version management and configuration management are not included in this.

A brief description is provided on the substance of each subject.

E.1 Management Summary

Provide a summary of the test plan here. Usual matters to include are:

- Reason for the project;
- Description of the project;
- Time lines;
- Costs, etc.

E.2 Introduction

What is this test plan about? Reflect the general structure of the document.

E.2.1 Documentation Used

Test Plan Documentation

Provide exhaustive reference to the source documentation used for this test plan, such as:

- Quick scan (inventory of the project using the Test Management Model);
- Planning;
- Test strategy;
- Risk analysis.

Bear in mind general documentation, such as:

- Project plan;
- QA plan;
- Configuration management plan;
- Relevant policies within the company.

The test plan should always refer to a higher level plan. If this template is used for a detailed test plan (e.g. Functional Acceptance Test) refer here to the project test plan.

Documentation for the System to be Tested

Provide a description of the "test basis" here. Think about things such as:

- Requirement specifications;
- Functional design;
- Technical design;
- User manual;
- Installation manual.

E.2.2 Standards and Procedures

Which standards and/or procedures will be used? Think about such standards as ISO 9126, ITIL, etc.

E.3 The Test Assignment

E.3.1 Client

Indicate who the client is.

E.3.2 Supplier

Indicate who, if applicable, the supplier is.

E.3.3 Goal of the Assignment

Provide a clear interpretation of the brief obtained from the client. Highlight the aim of the project and the result to be achieved.

The result to be achieved should be measurable. Establish agreement on how this is to be measured. The project should comply with the SMART principle: the assignment should be Specific, Measurable, Acceptable, Realistic and Timely.

E.3.4 Scope

Which test levels and tests are planned, what is explicitly *not* being tested and why *not*?

System Aspects to be Tested

Provide a description here of (the parts of) the information system, including the interfaces that are to be tested within the scope of the project. Bear in mind also the risks and importance of the system parts. These are defined in the test strategy.

System Aspects not to be Tested

Indicate what explicitly falls outside the scope of the project.

E.3.5 Suspension Criteria and Resumption Requirements

It may be the case that the documentation or software supplied is of poor quality or even incomplete. If so, there is little point in starting with the test analysis or test execution. Specify here, therefore, the criteria for the possible postponement of a part of the testing, or even the entire test. Make a connection to the entry criteria for the various test levels.

Also specify the test activities that should be repeated when testing is resumed following a period of postponement.

E.3.6 Test Project Deliverables

Which products do you intend to produce, and when? Provide a description and not a plan. Also indicate what the clients should do with the deliverables (for information, for approval) and *how* they should do it (they are not test experts).

Deliverables of a test project include:

- Test plan;
- Test specifications/analysis;
- Procedures;
- Test logs;
- Test issue report.

E.3.7 Discharging the Test

When is the testing complete? Who discharges it?

Also indicate the acceptance criteria that have to be met.

E.3.8 Starting Points and Preconditions

Indicate the conditions that must be met to allow the test project to succeed, or that have to be created to allow it to run smoothly. Be as explicit as possible.

E.3.9 Risks and Risk Countermeasures

These can be copied from the project risk analysis. Provide an overview here of the five (plus or minus two) project risks with the highest priority. Be sure to refer to the document with the complete risk analysis.

Plans often outline risks, without including risk countermeasures. Risks are also often mentioned that are really more tasks for the test manager and do not belong in the risk column. It is not necessary in a detailed test plan to describe the risks for which the test manager is responsible. However, in a project test plan, those risks should be described explicitly, in the first place to make the risks clear to the client (as part of prospect management) and secondly because the test manager may be responsible for preventive measures while someone else is responsible for corrective measures. The as-yet-unknown subjects in the cluster cards also form risks to the test project.

E.4 Test Approach

Provide a brief indication of what the quick scan has produced. Below are standard categories within a dynamic test approach. Risk- and requirement-based testing (RRBT) is one approach.

E.4.1 Test Strategy

Provide a brief summary here of the established test strategy. What is your approach; what have you considered, and why; what have you decided against, and why?

E.4.2 Preparation

What will you be doing during preparation of the various tests? Activities such as setting out the test management file, test environments, etc.

E.4.3 Analysis

For which parts of the information system will clusters, test conditions and test cases be made, and which not, and why not? Which testing techniques will be employed for the test specification and analysis? The choice of a particular technique depends on the quality attribute that is being tested, but also on the product risk. With a high risk, a more thorough technique will be employed.

You can refer here to the cluster cards to be set up, in which these subjects are to be covered.

E.4.4 Test Automation

If applicable, for which subjects will test automation scripts be made and for which not, and why not? Preferably refer to a cost–benefit analysis.

E.4.5 Execution and Transfer

What is to be addressed during the various test levels and who will do what? The information on this can be partly obtained from the test strategy.

Provide a brief overview of the various test levels with a description of:

- Where the testing will be done (environment);
- what will be tested;
- when it will be tested;
- who will be doing the testing;
- entry and exit criteria per test level;
- dependencies of other test levels.

Entry criteria may originate from before a test phase *or* elsewhere in the organisation (preconditions!).

Exit criteria may relate to a subsequent test phase or another part of the project (training, implementation).

Acceptance criteria are established in the test strategy. These criteria should be tested within the various test levels. The set with overall acceptance criteria should therefore be translated into entry and exit criteria for the test levels.

Remember, too, to carry out the evaluations, both during and at the end of the test project. When, as test manager, will you carry this out, who will you involve in it, and how will you do it? Will you, for example, use a measurement programme (GQM)? And will your whole project be concerned with this, or only a number of people?

E.5 Planning and Budget

Referring to:

- Test planning;
- resource planning (here or in the test organisation);
- time line planning (starting and finishing dates of the test phases);
- activity planning (Gantt chart in MS Project or EVM);
- milestone planning (brief summary of the test milestones);
- budget.

E.6 Test Organisation

E.6.1 Stakeholders

Who are the stakeholders; which parties play a role in the project (both internally and externally)? Derive an organisation scheme.

E.6.2 Tasks, Authorities and Responsibilities

Indicate per test role who is doing what, who takes which roles, and which tasks, authorities and responsibilities these concern. Possibly also provide information on cost and availability.

When describing the roles pay attention to the desired level of training. Also investigate the training possibilities if the desired level is not available.

When defining the resources, consider not only those in the test team, but also those whom you think you might need outside of the test project; for example, suppliers of the required hardware. It is important to survey all the relationships.

E.6.3 Meetings

When do meetings take place, with whom, about what, and how frequently?

E.6.4 Communication

What are you communicating about, and what agreements do you make with the client/project manager/project leader?

Bear in mind here:

- *Resources.* Who will take on the resources? Who will hold the intake interviews?
- *Reporting* (progress etc.). Who will set up the reporting, to whom will it be sent and with what frequency? If you use the test control matrix (TCM), refer to it.
- *Meetings.* See Section 6.3. To whom are the minutes sent per meeting?
- *Issue management.* Who will carry out the issue administration? Is this reported on, and to whom?
- *Quality plan.* Should a quality plan be written?
- *Training.* Should allowance be made for the training of users (both for the testing and for the implementation of the application)?
- *Escalation.* Who escalates to whom?
- *Reviews* etc. Are reviews to be held? Who will carry them out? Is this to be done in consultation with the client?

E.7 Description of the Test Environment

Hardware, software, tools (including any test tools); what is required to operate (should the environment simulate the production environment)? Much is already described in the test strategy and the various cluster cards, but it is possible to expand on this. Bear the following points in mind!

- Indicate who is responsible for setting up and managing the test environment.
- Indicate who is responsible for making available/ordering the infrastructure.
- Indicate whether you need, for example, production files.
- Describe also the required physical test environment (workstations, telephones, etc.).

E.8 Transfer of Testware to the Organisation

- Which party will transfer what to the recipient party?
- Are there acceptance criteria to be met?

E.9 Appendices

Add here the documents that have added value with the test plan:

- Quick scan;
- Test strategy;
- Planning;
- Risk analysis.

F

The Goal–Question–Metric (GQM) Method applied to testing

A new shoot on the tree of process and product improvement has appeared in the form of the goal–question–metric (GQM) method (Solingen & Berghout, 1999). In this case study we provide a brief description of the method, an explanation of the various steps that an organisation must take in applying it, and an example demonstrating how the application of the GQM method led to a number of clear improvements in an organisation.

Introduction

What exactly is the GQM method? Is it another means of allowing management to control employees from above? *No, not really.* Is it yet another way of documenting more wide-ranging procedures? *No, absolutely not.* Is it the umpteenth method based on the assumption that software testing is a straitjacket for every team and every individual? *No, not that either.* The GQM method is one that assists with the measurement and attainment of the most important goals within a test project. GQM is therefore *the* method for clarifying difficult, obscure situations by a pragmatic but theoretically sound approach. Clear to whom? To the client, to the project team, but especially to the testers themselves.

To demonstrate the kind of “difficult, obscure situations” referred to:

- How do you determine the quality of software components? When are software components suitable for release? What effective measures can you take to improve the quality of software components?
- How do you organise a test team efficiently? How do you determine the quality of test activities? How do you realise efficient reuse of a test?
- Where do software problems come from? What are the underlying causes of faults that are discovered during the test process? What can we do to prevent mistakes being made?

- What is the impact of interrupts in the work of a tester? Are there too many, or indeed too few, interrupts in an organisation? Why are testers disturbed so often, and can these disturbances be reduced?
- How effective are the various test levels? Are mistakes discovered in time?

The GQM method enables an organisation to perform measurements on a continuous basis, or for a fixed period, thus obtaining growing insight into the problems described. A GQM programme succeeds often, thanks to the strong integration, involvement and feedback on the part of development teams and testers. It is an excellent solution to the old problem of a quality organisation spending much time and effort collecting more or less unusable measurement data.

The GQM method has been very successfully applied and further refined by, among others: NASA, Hewlett Packard, Motorola, Schlumberger, Ericsson, Digital, Nokia, Philips, Tokheim, Dräger, Robert Bosch, Daimler Chrysler, Siemens, Allianz and Proctor & Gamble.

The GQM Method

The GQM method is based on the principle of targeted measurement. "To measure is to know" is a well-known Dutch saying, but only holds true if you know what exactly you need to measure. With this method, you start by formulating the Goal of the measurement programme, then you document the relevant Questions to support your goal, and subsequently you determine the Metrics to be set up within the organisation in order to answer the questions. Also, for each question and metric a hypothesis is put to the test team. Comparing real values from the measurement application programme with these hypotheses provides the testers with better insight into their own work. GQM is therefore more than anything else an approach that helps testers to learn from their own work and experiences.

Since GQM specifies the metrics through goals and questions, it is not only an approach for defining metrics, but also for interpreting measurements. The measurements serve to answer the questions, and when all the questions have been answered, it is clear whether the goal has been attained. GQM focuses not only on the specifying of goals, questions and metrics, but above all on providing answers and attaining goals. This goal-oriented aspect of GQM makes the approach suitable for application to (goal-oriented) company environments.

There are four phases of a GQM measurement application programme (see Fig. F.1):

- *Planning.* Selecting the goals of the measurement application programme. The goals are geared towards the specific context of the organisation: what does the organisation want to know, what is the measurement application programme required to achieve?

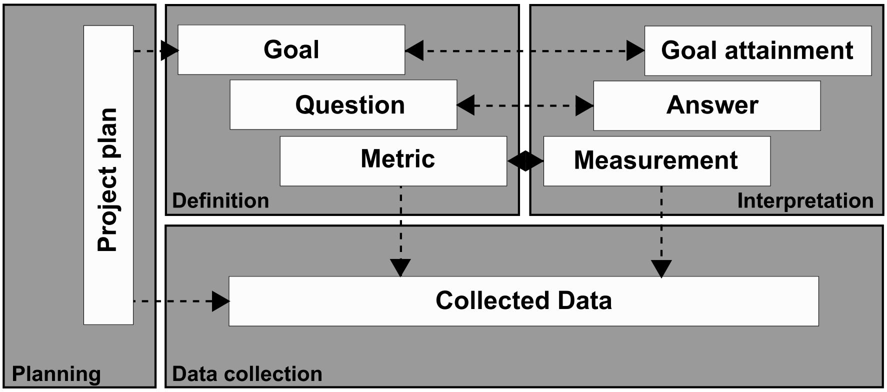

Fig. F.1. The GQM method

- *Definition.* Defining the goals in terms of measurement application, questions and metrics. As with establishing the goals, this is done in close consultation with all the stakeholders in the organisation, the test teams, developers and the testers. Through various interviews, the measurement application programme is defined and documented. In many cases, a process description is also created relating to the relevant test processes being measured (such as the acceptance test process or the performance test process).
- *Data collection.* The collection of data in accordance with the documented measurement application plan. All the testers and developers collect data manually or automatically, and the data are stored centrally.
- *Interpretation.* Presenting the data within the organisation for analysis, for drawing conclusions, taking decisions and proposing measures. The interpretation sessions take place at regular intervals and are the part of the measurement application programme that yields the biggest results. Through intensive feedback with test teams, more and more insight is gained into the problems of the organisation, and improvement proposals are presented which are carried through into the whole of the organisation.

The *interpretation* phase, apart from being the most interesting one, is also the most essential phase of the programme. This is when conclusions are drawn, decisions taken and actions defined. If measurement data are not analysed and interpreted, all the money, time and energy expended represent a waste of precious resources. In feedback sessions, learning is acquired and improvements defined. That feedback sessions are so crucial to the success of a measurement application programme is one of the most important findings of this research in recent years. Absence of the interpretation phase, or carrying it out too late, is the biggest cause of failure of measurement application programmes.

In Solingen and Berghout's GQM book, a list of activities and products is provided for each of these four phases. There is also a checklist that may be used at the conclusion of a phase to ascertain whether any important points have been overlooked.

The GQM approach is based on the paradigm of Victor Basili and David Weiss (see Fig. F.2). GQM is based on the assumption that organisations wishing to improve should first of all document their improvement goals. On the basis of these goals, a course is set whereby metrics are applied in support. It is therefore essential to establish in advance which information and learning requirements exist in the organisation. Quantitative information is collected to support this process.

Since the questions and metrics are defined with an explicit goal in mind, this information must only be used to interpret along the lines of the goal. We should be careful not to yield to the temptation of drawing conclusions beyond the measurement goals. The reason for this is that the collected data are very probably insufficient to draw such conclusions; there is a good chance that important aspects have been left out, and the conclusions might be entirely mistaken. The end result of the application of GQM is a measurement application environment focused on a set of specific improvement goals within an organisation.

Distinction should be made between "improvement goals" and "measurement goals". Improvement goals indicate the degree and direction required in improving performance. For example, a productivity increase of 30%, or a timetable reduction of 50%. Measurement goals, on the other hand, only specify an information requirement with a reason and a focus. As such, measurement goals do not deliver improvements, but they do deliver better insight. With the aid of this insight, of course, measures can be taken to achieve

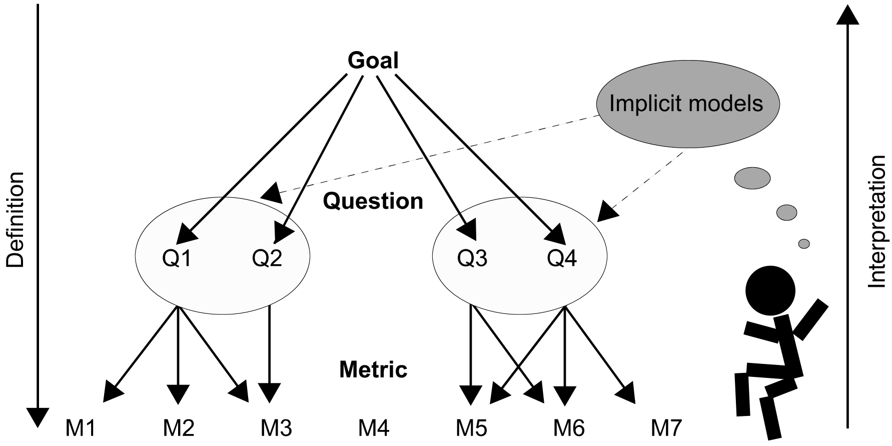

Fig. F.2. The GQM paradigm of Basili and Weiss

the improvement goals. However, the difference between the goals is essential. GQM assists organisations to acquire specific information and explicitly demands interpretation of this information. GQM thus provides an operational learning process. In interpreting the data, questions relating to the "how" and "why" of significant phenomena are analysed and argued. To support this learning process, it is important that an expected answer to each question is documented in advance. Such hypotheses demonstrate explicitly what people expect on the basis of their current thought processes and experiences. Final results can be compared against these expected answers and differences analysed. In this way, GQM stimulates focused learning and links it to people's current knowledge, a process that fits in very well with the way in which adults appear to learn.

Measuring in Practice

If we now look at how measuring is done in practice, we quickly become disillusioned. Only a very few organisations apply structured measurement to test and development activities. Many reasons are given, the most usual of which is that it takes a lot of time. However, research has shown that the input of the testers need be only very limited, provided they are supported appropriately. Calculations of the return on investment (ROI) of measurement application programmes are always positive, and there are situations where ROIs have been measured at 35 or higher. This is logical, when we look at the costs of a problem that is discovered after release. Only a small number of extra problems need be found through a measurement programme for it to end in a positive result. If we look in particular at the market for embedded systems, the ROI is much higher, since changes to embedded systems can often take place only when the manufacturer takes a series of products back to the factory. The direct costs of this (apart from the damage done to the company's image) are so great that such companies very quickly recoup the extra resources spent on finding or preventing problems.

The right way to implement measurement application programmes in practice is with the help of a GQM team. This team supports the testers within a measurement application programme by taking over all those activities that do not necessarily have to be carried out by testers. For example, the writing of a GQM plan, designing data collection forms, preparing a feedback session, collecting measurement data and setting up graphics and tables. Someone outside of the test team can carry out such things perfectly well, so that the normal test activities are put under as little pressure as possible. It is important that this GQM team does not interpret the data – this is a task for the testers themselves!

The analysing of measurement data takes place during feedback sessions. In these sessions, all the parties involved come together and take a collective look at the collective data. During interpretation sessions, conclusions are drawn, decisions taken and actions defined. Analysis is done with the help of the GQM questions. For each question, a number of overviews of the relevant measurement data are indicated and it is examined whether the question can be answered. It often turns out that parties have completely different ideas on correlations and causes. The discussions that ensue stimulate the learning processes of everyone concerned. That is why interpretation sessions represent one of the critical success factors for measurement application programmes in practice.

Example of a Measurement Application Programme Within a Test Department

In the following case, a measurement application programme is described, which was carried out in an industrial organisation. The measurement application goal of the programme was:

Analyse:	The system testing process
With the aim of:	Understanding
In relation to:	Effectiveness and efficiency
From the point of view of:	The system testers

Based on the input of the testers, this measurement application goal was represented as the following list of questions:

1. What is the degree to which conditions for satisfactory testing are met in the current projects?
2. Which factors influence the costs and duration of the system-testing process?
3. What is the contribution of system testing to product quality?
4. What determines the duration of the system-testing process?
5. What are possible indicators and standards on which decisions on halting system testing can be based?

Based on these questions, a list of metrics was set up and data collected over a period of 2 years. In Figs. F.3 and F.4, two examples are provided of measurements that help to answer the questions above. These measurements can also help with decision making during testing, or aid in planning and estimating tests.

Figure F.3 shows the status within project A of the found, open and closed issues. With the help of this overview, developers can plan time for correcting faults. This diagram also shows that it is not so much the number of found issues that indicate quality, but more the number of open issues remaining. This is the number of issues that may confront the client (apart, of course, from the issues not yet found).

Apart from these metrics, the average duration of an issue was also documented, as well as the average time it took to solve an issue. In this way, test

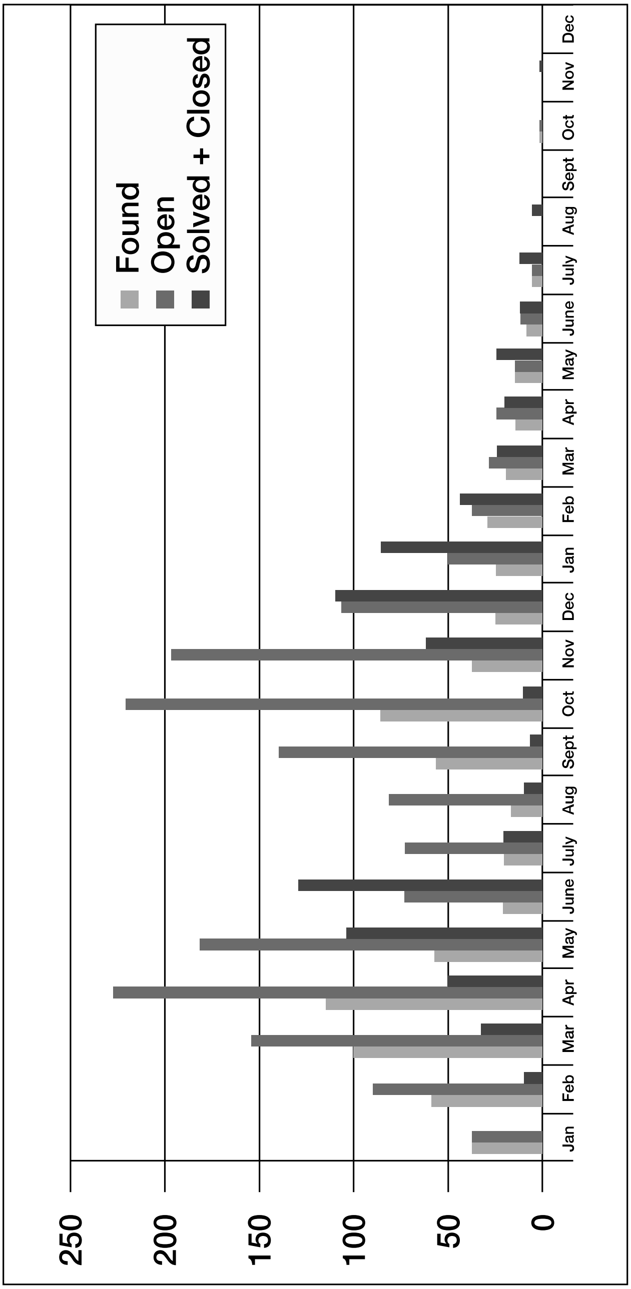

Fig. F.3. Project A: Fault status overview

activities and problem solving can be better coordinated and planned. This is an essential product of measurement application programmes: organisations obtain data on their own performance. It is therefore possible not only to see where improvement is required, but also to see the "personal records". This makes it possible to estimate directly how realistic the set plans and budgets are.

Besides an overview of faults, Fig. F.3 also provides insight into the effectiveness of test techniques. The manner of testing is related to the result. With the help of this figure, it can also be indicated during which period the most success was achieved in finding issues and which techniques were used there. It is notable that the summer period (July–August) shows a decline. This is of course a result not of the techniques used, but the reduced effort owing to holidays.

It is essential that the testers themselves interpret such statistics as shown in Fig. F.3. They know exactly what has been happening in a particular period. There are always influencing factors for which no metrics are collected, and conclusions concerning what exactly the data demonstrate and what action is required can only be taken by the testers themselves. These conclusions are discussed during interpretation sessions, where the most important learning effects take place.

Figure F.4 shows an overview of the subsystems within the project and the conditions required for appropriate testing. The testers have indicated

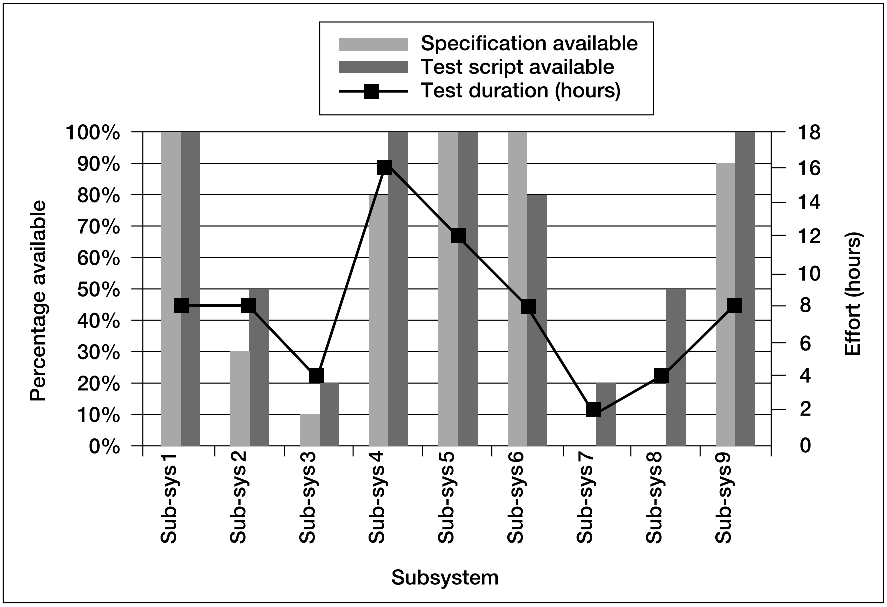

Fig. F.4. Status overview of test conditions per subsystem

for each subsystem to what percentage the required input (specification of the subsystem and test script for the subsystem) is available. Using this, they were able to show that the testing could be carried out much more effectively if such conditions were well organised.

As can be seen from this diagram, for a number of subsystems there is only a limited amount of documentation available and also a limited amount of test scripts. In this way the possible effectiveness of the test process can be estimated, as well as what needs to be done to improve it. The right-hand side of the diagram shows the test effort per subsystem. This shows that for the subsystems with limited compliance with sub-conditions, less time is also required for testing, which is an indicator of effectiveness (note that this relates to system testing!). If we were to show the test coverage in this diagram, the exact opposite would be seen. Subsystems for which test scripts and specifications are available are more thoroughly tested. Furthermore, system tests can be planned and estimated more effectively with these figures.

The above examples demonstrate that metrics can help us to learn about average performances and what can be done to improve them. To measure is to know, and without measuring it is difficult to obtain such exact figures. In test projects especially, it is essential to take measurements. First, because test projects are usually subject to pressures of time and there is a need to work effectively and efficiently. To do so, precise coordination is required and statistics are therefore necessary. Secondly, testing is a discipline in itself and testers should have the opportunity of learning from their own experience. Finally, it is the case that the costs of testing have increased dramatically over recent years in many organisations. More focused testing is necessary to stem these costs. Applying measurements helps in mapping costs and determining which test activities are more or less effective than others.

G

Checklist of Project Risks

This checklist contains a questionnaire regarding possible project risks.

It is subdivided into the following categories:

- Organisation;
- test project;
- test team;
- the selected test method;
- the information system;
- test environment(s);
- automated testing.

Each question is followed by a selection. The bold answer results in the highest project risk.

For example:

No.	Project risk	Selection	Importance (H/M/L)
3.1	Has a project leader been assigned to the test project?	• Yes, full time • Yes, part-time • **No**	

This example shows that the highest project risk occurs when no project leader is assigned to the test project.

G.1 Organisation

No.	Project risk	Selection	Importance (H/M/L)
1.1	Is a clear test organisation defined?	• Yes • **No**	
1.2	Is the project control from the client clearly covered and high enough in the organisation?	• Yes • **No**	
1.3	Are there many changes in the organisation structure? Many changes in the structure of the customer organisation can delay the project	• Stable, remains stable • Unstable, becomes stable • **Unstable, remains unstable during the project**	
1.4	The culture of the customer organisation can be characterized as:	• Open culture, healthy degree of ambition • Average • **Closed, uncooperative, conservative**	

1.5	How can the organisation be characterised (e.g. standards/ procedures)?	• Flexible, quick and strong drive for results. • Pretty formal, many consultation meetings • **Strong hierarchy, long communication lines, decision making highly political**
1.6	Are the (future) users able to cope with changes, or is there resistance?	• High • Medium • **Low**
1.7	Is tooling available for version control of the testware?	• Yes and known • Yes but no experience with the tooling in the test team • **No**
1.8	Are standards and procedures stable and sufficient (document templates, user interfaces etc.)?	• Yes • Yes, not sufficient • **No**
1.9	Are the organisation's expectations regarding the test project realistic?	• Yes • **No**

G.2 Test Project

No.	Project risk	Selection	Importance (H/M/L)
2.1	Have all the test project's conditions been defined and approved, regarding, for example, test environment and tools?	• Yes • Not yet, but this action is part of the starting points of the test project • **No, and this action is *not* part of the starting points of the test project**	
2.2	The scope of the test project can be the extension of an existing test set or the creation of a new test set. The projects scope is:	• Maintenance • Extension of testware • **Development of new testware**	
2.3	What is the estimated size of the test project in working days (size only refers here to test employees)? A working day is based on the amount of work one person with 6 months' test experience can do per day	• Less than 125 days • Between 125 and 350 days • **More than 350 days**	
2.4	The estimated size of the test project expressed in working days is:	• Less than 30 days • Between 30 and 90 days • **More than 90 days**	

2.5	On how many other projects does the test project depend? It can become complex when the test project is dependent on deliveries of other projects.	• No dependencies with other projects • 1 or 2 • **More than 3**
2.6	Can the test project be divided in time boxes (e.g. RAD) where the systems functions are developed in logical coherent and self-containing units? If the project consists of exactly one time box: No	• Yes • Most are • **No**
2.7	Are time boxes that are executed in parallel autonomous? If the project consists of exactly one time box: yes	• Yes • Most are • **No**
2.8	The project consists of how many time boxes? The number of independent time boxes gives the same number of independent components. This implies a consistent development process (for users and developers), integrating components, etc.	• 1 to 5 • 6 to 10 • **More than 10**

2.9	Is technical support included in the project budget? The development team can give input to the test team about technical details of the development environment and other technical details necessary for the test preparation and execution	• 20% or more budget (human resources) available for support • 5 to 20% budget (human resources) available for support • **No budget for technical support**
2.10	Domain experts are essential on the test project, especially when the requirements of the application under evaluation are complex and are poorly or not documented. How do analysts relate to domain experts?	• Average 3 analysts to 5 domain experts • Average 1 analyst to 4 domain experts • **No domain experts available**
2.11	How high is the work pressure for domain experts in their own organisation? This can affect participation of the business experts in the project	• Low • **High**

2.12	Do the domain experts have a mandate to make decisions about the scope of the test? Quick-wittedness, independence, and authority are critical success factors. The mandate must have sufficient support by the management and employees of the committed departments	• Yes • **No**
2.13	When the end user organisation consists of several departments/stakeholders, does this mean that possible opposite interests must be overcome? The end user organisation consists of:	• 1 department • 2 departments • **More than 2 departments**
2.14	Is a procedure defined for approving the testware? Is it clear how the test will be approved with analysts, users and management?	• Yes • **No**
2.15	Are acceptance criteria defined?	• Yes • **No**

2.16	Is it clear how to deal with changes in starting points, functional specifications, requirements, standards, etc.? Is a change control procedure defined and is this procedure connected to the change control procedure of the system under evaluation?	• Yes, connected to the systems change control procedure • No, independent of system • **No**
2.17	Is enough testware review capacity available? Only applicable when this capacity is not in the projects scope	• Yes • **No**
2.18	Is it clear to the client that part of he test effort can only start after delivery of the test object?	• Yes, and clearly part of planning • Sufficient time has been planned for this • Little time has been planned between delivery and start of test execution • **No, fixed planning without dependencies between moment of delivery and start of test execution**
2.19	How many geographical locations does the project have?	• 1 location • 2 to 3 locations • **more than 3 locations**

2.20	Is sufficient support available from, for example, a support department?	• Yes • **No**	

G.3 Test Team

No.	Project risk	Selection	Importance (H/M/L)
3.1	Does the test project have a test manager (TM)?	• Yes, full time • Yes, part-time • **No**	
3.2	Does the TM have experience with test projects?	• TM has been TM before in other test projects • TM has test analysis or test team leading experience • **TM has no test experience**	
3.3	Are the analysts and navigators (test team) capable of working in an independent and structured way?	• All test team members can work independently • Most test team members can work independently • **No, or only a few test team members can work independently**	
3.4	What is the size of the test team?	• 1 or 2 • 3 to 5 • **More than 5**	

3.5	Do the members of the test team have experience with testing?	• All members were previously involved in test projects • Most of the members were previously involved in test projects • **None or some members were previously involved in test projects**
3.6	Are the test team members motivated to participate on the project?	• Yes • Yes, but not their main task • **No**
3.7	To what extent do the project members know the clients organisation?	• All required knowledge • Sufficient knowledge • **Insufficient**
3.8	What will the turnover of staff be during the project?	• Nil • Limited • **Large**
3.9	To what extent can domain experts work together with testers, in terms of open communication, frequency, intensity and physical location?	• Good • Reasonable, if certain conditions are met • **Bad**
3.10	Are enough employees available for test preparation, design and execution?	• Yes • **No**

G.4 The Selected Test Method

No.	Project risk	Selection	Importance (H/M/L)
4.1	If the test method is new to the organisation, does the TM have organisational skills to embed new methods and technologies and to manage the projects environment?	• Yes • **No**	
4.2	Do all testers have knowledge of the chosen test method?	• All testers were previously involved in test projects with the chosen method • Most of the testers were previously involved in test projects with the chosen method • **None or some testers were previously involved in test projects with the chosen method**	
4.3	The choice for adopting the test method is ...	• Explicitly made by the client • Made in consultation with the client • **Implicitly made**	
4.4	The customer organisation is willing and ready to accept the chosen test method	• Yes • **No**	

4.5	Is the customer organisation capable of implementing the new test method?	• Yes in a mature way • **Hardly: too much innovative enthusiasm or ungrounded conservativeness**	

G.5 The Information System

No.	**Project risk**	**Selection**	**Importance (H/M/L)**
5.1	The impact of defects in the system increases with the number of concurrent users of the system. The system must support:	• 0 to 5 concurrent users • 6 to 30 concurrent users • **More than 30 concurrent users**	
5.2	How many types of users have to work with the system under test? Also keep in mind supporting users like system administrators, maintenance personnel, etc.	• 0 to 3 types of users • 4 to 6 types of users • **Over 6 types of users**	
5.3	Are domain experts capable of separating essentials from side issues? Is the user capable in weighting the requirements in terms of being required, desirable or optional?	• Yes • **No**	

5.4	The development and test activities of a system that supports the core processes of an organisation will get more priority and a greater budget than a secondary system. The system is a ...	• Primary system • Subsystem of a primary system • **Secondary system**
5.5	Systems that support the core processes of an organisation are often more complex and therefore more effort is required for test activities. The system is a ...	• Secondary system • Subsystem of a primary system • **Primary system**
5.6	In some cases, the repair of defects in or downtime of the system in the production environment is very expensive. Other systems are replaced relatively easy by, for example, manual procedures. The system not functioning under test is ...	• Relatively easy to cope with • Very expensive • **Not possible**
5.7	When the changes that are implemented in the system under test are fewer with respect to the previous release, the effort of the test activities can be reduced. The effort of the test activities can also be reduced when a working version of the system is available during the development of the test set. The system is ...	• Stable and available • Unstable and available • **New, under construction**

5.8	Is it easy to use and control the system s functionality or is extra tooling required?	• Data input and validation during testing is easy • Data input and validation is difficult (e.g. several screens and files must be processed) • **Additional tools are required for data input and/or validation**
5.9	Are the systems functions clearly documented and is this documentation available?	• Yes • Partially • **No**
5.10	Is there sufficient documentation about the system available?	• Yes • **No**
5.11	Is the system to be executed on 1 or several platforms?	• 1 • 2 • **3 of more**

G.6 Test Environment(s)

No.	Project risk	Selection	Importance (H/M/L)
6.1	Is a separate, dedicated environment available for testing?	• Yes • **No**	
6.2	In what way is the test environment independent of the development environment?	• Independent, standalone • Is partly using the system • **Totally integrated**	
6.3	Is the test environment always available for each test?	• Yes • **No**	
6.4	Are test runs influencing the production environment?	• No • Limited • **Yes**	

6.5	How many technical environments are in the test scope? Number of hardware/software platforms, e.g. DOS/Windows with Power Builder and C++ (= 2 environments), a Sybase Database server and an AS/400 system (= 2 environments), usage of a Sybase database server and an IBM mainframe	• 1 • 2 • **3 or more**
6.6	Number of interfaces with other systems? How many other systems (not in the project scope) must be communicated with?	• 0 • 1 to 2 • **3 or more**
6.7	Is the test environment in line with the production environment? Included are hardware and software, e.g. amount of internal memory, type of display, network, middleware, servers	• Completely identical • Almost identical • **Strongly diverse**
6.8	Is a test development environment available and is it solely available for the project?	• Yes, available for the project only • Yes, but must be shared with other projects • **No**

6.9	Is the test environment completely arranged?	• **Completely** • **Partially** • **No**
6.10	To what extent is the test environment well defined? For example: • All subsystems will be tested or not. • Every field will be tested or not on characteristics. • All functionality within a subsystem will be tested, or only the most important	• Determined, well defined • There is no fine distinction • **Variable, but hardly any knowledge about the distinctions**
6.11	Does a department exist that has control of the test environment as core business?	• Yes • **No**

G.7 Automated Testing

No.	Project risk	Selection	Importance (H/M/L)
7.1	Will automated testing be used on the project? If not, the rest of the questions are irrelevant	• No • **Yes**	
7.2	Are the testers experienced with automated testing?	• All testers have experience with automated testing • Not all testers have experience with automated testing, but they have programming experience • **None or some testers have experience with automated testing or programming experience**	
7.3	Do the testers have knowledge of and experience with the environment of the system under test?	• All • Some • **None**	

7.4	Are the testers experienced with the selected test tool?	• All testers are experienced with the tool • **One or some testers are experienced with a tool, but all followed the training course**
7.5	A test project can support several goals, e.g. to measure if the quality of a version of a system is sufficient or to improve the efficiency of the test process (improved speed over lower cost). Is the goal of the test project the improvement of the test process's efficiency?	• Yes • **No**
7.6	Is the goal of the test project the reduction of test personnel?	• Yes • **No**
7.7	Is the client aware that a short-term investment is required and that this investment will only have long-term results?	• Yes • More or less • **No**
7.8	Is the client experiencing external pressure for the development of repeatable regression tests? For example, auditors, accountants	• Yes • **No**

H

Template for Test Strategy

This template is a tool to develop a test strategy. Analogously to Chapter 5, this template describes seven steps to draw up the test strategy. Also, intermediary products are delivered. In the final version of this document one can choose to incorporate only the final products (product risks, acceptance criteria, cluster matrix and cluster cards). However, the intermediary results should be kept as proof of how the final products were put together.

General aspects such as version management and configuration management are not included in this template.

A brief description is provided on the substance of each subject.

H.1 Management Summary

This paragraph must contain a summary of the test strategy, which describes the most important issues of the tests. Additionally it must describe what is tested and what is not tested in the test project.

This summary must include the cluster matrix.

H.2 Introduction

The test strategy is a way of communication between client, stakeholders and the test team. This document indicates what will and what will not be part of the test project. For this the input of the stakeholders is necessary.

H.2.1 Seven Steps

To draw up a test strategy the following steps must be undertaken:

- *Step 1: Identify the stakeholders* – Who are the parties concerned with the test project, and who is responsible for accepting the various tests and should therefore be involved in the test project?

- *Step 2: Carry out the test product risk analysis* – What are the risks that are directly related to the information system to be tested? What should be tested in order to meet the requirements and wishes of the stakeholders?
- *Step 3: Link the relevant product risks to the quality attributes* – Which product risks can be linked to which quality attributes?
- *Step 4: Determine the test levels* – What are the best ways of testing the various product risks and quality attributes?
- *Step 5: Determine the acceptance criteria* – At which point do the client and stakeholders find the quality of an information system acceptable? What acceptance criteria do they apply?
- *Step 6: Set up the cluster matrix* – Group the tests so that logical clusters are created within which exactly one stakeholder is taking responsibility.
- *Step 7: Set up the cluster cards* – A cluster card contains all the information that is of importance to the testers in setting up the tests for a specific cluster. The test strategy is clearly set out in the cards, showing all the parties involved. The cluster cards form the end of the test strategy and the beginning of the test analysis.

Further, a short description of the test project should be given in this introduction. This description must at least contain the following issues:

- The test object;
- stakeholders identified within the quick scan;
- whether the stakeholders will be assigned to the test level, test environment, department or parts of the information system, including the arguments on which these decisions are based;
- the way in which the risks in the cluster cards are derived.

H.2.2 Documentation Used

Give a full reference to the source documentation used for this test strategy, such as:

- The quick scan;
- the project risk analysis.

H.3 Stakeholders and Organisational Structure

Stakeholders are functionaries or departments that have a direct interest in a properly working information system.

During the quick scan most stakeholders were identified. These stakeholders can be described in this paragraph. To identify the missing stakeholders the following question can be asked: “Who (which department) is responsible for (parts of) the information system?”

H.3.1 Stakeholders of the System

Give in this paragraph an overview of the stakeholders. In general the stakeholders can be categorised in the following way:

- End users;
- marketing department;
- supporting departments, such as the helpdesk;
- IT department;
- internal control/audit.

H.3.2 Organisational structure

Describe here the organisational structure in an organisation chart.

If the organisation chart is also used in the quick scan, it can be reused here. Make sure the stakeholders can be identified in this chart so their place in the organisation is clear.

H.4 Link product risks and quality attributes

In this paragraph, the results of Steps 2 and 3 will be described.

H.4.1 Product risk analysis

Define together with the stakeholders the most important product risks.

The product risk checklist can be used to identify these risks. The risks that where distinguished in the quick scan can also be used here.

The identified product risks should be transferred to a table like that shown below:

Product risk	Priority (MoSCoW)	Stakeholder
...	...	...

H.4.2 Link product risks and quality attributes

In this paragraph the product risks will be grouped into the ISO 9126 quality attributes.

This link will facilitate communication with the client and the stakeholders. It will also offer an opportunity to check if the product risks are complete. When there is no link between the risks and a quality attribute, this could mean that some risks are missing. These should be added. It could also mean that the quality attribute in question does not have to be tested. Make an explicit remark for this!

Product risk	Quality attribute	Priority (MoSCoW)	Stakeholder
...	...	...	...

H.5 Test levels and acceptance criteria

H.5.1 Test levels

In this paragraph, the different test levels in the test project will be described. Also give a definition and a description of the test levels to avoid any misunderstandings at a later stage.

Possible test levels are:

- Unit test;
- integration test;
- system test;
- user acceptance test;
- production acceptance test.

H.5.2 Acceptance criteria

The stakeholders will define acceptance criteria for the information system. If the information system meets these predefined criteria, the test team is finished with testing and the product risks are reduced to an acceptable level to the stakeholders.

Acceptance criteria can be defined in many different ways. It is important that an acceptance criterion is specific and measurable. Think of the SMART rules: Specific, Measurable, Acceptable, Realistic and Time driven. The acceptance criteria will be translated in the test plan to entry and exit criteria for the different test levels.

H.6 Cluster matrix

A cluster matrix is an overview in which the test manager will clearly group the stakeholders, the quality attributes and the different test levels. It is then easy to see who is responsible for what quality attributes and at which level they will be tested.

To define the best test levels for the quality attributes, the following questions can be asked:

- Which quality attribute can be tested at what test level?
- Are there quality attributes that can be tested together at one test level?
- Which preconditions apply when performing a test for each specific quality attribute?
- To minimise costs (the earlier a bug is found, the cheaper these are) what is the earliest stage to test a quality attribute?

The above steps result in the cluster matrix. See the table below for an example where the quality attributes are linked to product risks, stakeholders and test levels.

Stakeholder/ test level	End user	Marketing	Department	IT department	Internal control
Unit test					
Integration test					
System test				• Recoverability • Analysability • Changeability • Interoperability	• Accuracy
User acceptance test	• Suitability • Understandability • Learnability				
Product acceptance test				Time behaviour	

H.7 Cluster cards

The first (main) clusters are defined in the cluster matrix. Then the test manager will look at the relative importance of the different quality attributes within this first clustering. The importance of this is derived from the defined product risks. The more and the higher the product risks, the higher the priority. If all the quality attributes in one cell have the same priority, then this one cell can be directly translated into one cluster card. If one priority of a quality attribute is different, then more cluster cards must be defined. This is necessary, because the test conditions and test cases will have the same priority as the cluster.

A cluster card contains information divided into four categories:

- key information;
- assignment;
- execution;
- result.

Key information	
Cluster name	Logical cluster name
Information system	System name and version
Test level	Within which test level will this cluster be included?
Test department	Which department will test this cluster?
Stakeholder	Who has an interest in, and responsibility for, the correct working of this cluster?
Assignment	

Product risk(s)	Which product risks could materialise if this cluster is not processed, or if faults appear in this part during production?
Requirement(s)	Which requirements are linked to the product risks covered in this cluster? Testing should prove that the requirements are met (proven added value to the organisation)
Priority	What is the importance of this cluster? The importance corresponds to the priority of the product risks, and influences the sequence in which the test manager will plan the clusters. Use the MoSCoW classification: “Must test”, “Should test”, “Could test” and “Won’t test”. By inferring the cluster importance from the product risks and their importance, we prevent the stakeholders from allocating the highest importance to all the clusters
Quality attribute	Which quality attributes underlie this cluster?
Source material	A reference should be included here to the requirements, the documentation and interview reports upon which the design of the test will be based
Execution	
Test approach	How will the test be executed? The test manager’s choice depends on, among other things, the test level, the quality attribute to be tested, the available source material, the organisation and the circumstances A further breakdown is created into: • Static testing: auditing and reviewing • Dynamic testing: testing the application itself Which test techniques will be used: decision tables, entity life cycle, data flow analysis, etc.? A choice is also made here between manual or automated test execution
Test environment	What test environment is required to be able to carry out the tests described? Both the technical environment and the necessary resources and time-dependent aspects should be mentioned here Which test data will be used?

Results	
Acceptance criteria	At which point will the stakeholder accept the cluster? Ensure that the acceptance criteria are set out explicitly. All the parties involved should know in advance when the test project will be complete

At this stage of the project, it is not a major problem if the test manager is not able to fill in all the fields in a cluster card. Many of the fields will be filled in while the test team is starting with the design of the tests. Until these empty spots are filled, they are project risks or actions for the test manager. He must guard against the risks.

The cluster card is the first tab in an Excel cluster for analysis. To avoid doing things twice it is best to use and fill this Excel workbook. The test manager can refer to this in the test strategy.

I

Testing Roles

Within organisations it is important to have formally described test functions. By developing a career path in this way an organisation avoids disintegrated test knowledge and expertise and it gives employees the possibility to specialise in testing. Of course this should include education and coaching. This appendix will describe possible test roles: test manager, test team leader, test analyst, tester, test navigator and test consultant.

TEST MANAGER

Function description

The test manager manages the test team. He is responsible for the budgeting, planning and organisation of all test activities: drawing up test plans, progress reports and final reports, directing and executing the test process. In addition, he is responsible for reporting on the quality of the “test object”. The scope of the test manager is broader than the scope of the test coordinator. The test manager leads bigger test teams and the focus is on the entire project or multiple projects

Tasks, responsibilities and authorities

- Reaching agreement with the project manager/client about the test assignment and expectations
- Ensuring a test infrastructure is available on time
- Keeping the issue administration continually updated
- Communication on all test issues within the project team
- Transfer of all testware to the maintaining party after completion of the project
- Allowing use of all agreed test methodologies
- Creation and maintenance of project test plans and detailed test plans
- Overseeing the planning, budget and execution of the test process
- Reporting on progress of the test process
- Reporting on the quality of the test object

General knowledge	• Higher vocational education or academic work and thought level, study related to ICT • ICT - knowledge – familiar with development methods (SDM, DSDM, RAD, CBD) – knowledge of system design techniques (functional design and technical design) – knowledge of ITIL processes – experience with project management methods – experience with test support tools – familiar with quality assurance – familiar with risk management – familiar with prototyping
Technical knowledge	• Detailed knowledge of and experience with test methods, test techniques and test support tools • Practical experience in the use of planning and budgeting techniques and support tools • Practical experience in providing leadership to the different test roles • Knowledge of the application of test process improvement models (TMM, TPI) • Knowledge of moderating group inspections (Fagan inspections) • Knowledge of the execution of test plan audits • Knowledge of the automated test concepts • Knowledge of the use of automated test support tools

Skills	• Flexibility • Group-directed leadership • Individual directed leadership and delegation • Persuasive skills • Communication skills • Planning and organization • Problem analysis and judgement forming • Performance motivation • Drive and self-confidence • Ability to listen and be sensitive to circumstances • Sensitivity to organisation in which operating • Takes initiative • Pragmatic approach
Work experience	• Experience as a tester, test analyst and optionally as test navigator • At least 1 ear’s experience as test coordinator • Experience of project-Related leadership (minimum 2 years)

TEST TEAM LEADER

Function description

The test team leader manages the test team. He is responsible for the budgeting, planning and organisation of all test activities: drawing up test plans, progress reports and final reports, directing and executing the test process. In addition, he is responsible for reporting on the quality of the "test object". The scope of the test team leader is narrower than that of the test manager. The test team leader leads smaller test teams and the focus is on one (part of a) project

Tasks, responsibilities and authorities

- Reaching agreement with the project leader/client about the test assignment and expectations
- Ensuring a test infrastructure is available in time
- Communication about all test matters within the project team
- Transfer of testware to the maintaining party after completion of the project
- Allowing use of all agreed test methodologies
- Creation and maintenance of test plans
- Overseeing the planning, budget and execution of the test process
- Reporting on the progress of the test process
- Reporting on the quality of the test object

General knowledge	• Higher vocational education or academic work and thought level, study related to ICT • ICT knowledge: – familiar with development methods (SDM, DSDM, RAD, CBD) – knowledge of system design techniques (functional design and technical design) – knowledge of ITIL processes – experienced with project management methods – experienced with test support tools – familiar with quality assurance – familiar with risk management – familiar with prototyping
Technical knowledge	• Detailed knowledge of and experience with test methods, test techniques and test support tools • Practical experience in the use of planning and budgeting techniques and support tools • Practical experience in providing leadership to the different test roles • Experience with the application of test process improvement models (TMM, TPI) • Experience with moderating group inspections (Fagan inspections)

	• Familiar with the execution of test plan audits • Familiar with the automated test concepts • Familiar with the use of automated test support tools
Skills	• Flexibility • Group-directed leadership • Individual directed leadership and delegation • Persuasive skills • Communication skills • Planning and organisation • Problem analysis and judgement forming • Performance motivation • Drive and self-confidence • Ability to listen and to be sensitive to circumstances • Sensitive to the organisation in which he is operating • Takes initiative • Pragmatic approach
Work experience	• Experienced as a tester, test analyst and optionally as test navigator • Experienced with project-related leadership (minimum 2 years)

TEST ANALYST	
Function description	The test analyst is responsible for developing the test analysis. In order to determine and define the test conditions and test cases based upon the product risks and requirements, he uses test specification techniques. The test design forms the basis for the test scenario, in which the test analyst defines the execution order of the test cases. In addition, he is charged with the capture of test results and the use of test support methods (as part of the automated test execution). The test analyst uses methods, techniques and support tools to do the job. He also helps with the checking of design documentation. He registers all findings relating to anomalies that arise between the information system and/or the system design and/or the test design
Tasks, responsibilities and authorities	Intake test basis: • Evaluate delivered design and specifications for testability (in the test basis) • Report on the established quality of the test basis • Analysis of the design and specifications • Design and documentation of test cases with relevant state situations, result expectations and execution instructions

	• Recording the test results and the necessary test support tools (as part of the automated test execution) • Maintaining the testware within the project: for both manual and automated testing • Transfer of all testware to the maintaining party after project completion • Providing support to testers and test automators during test execution • Reporting to the test leaders • Making use of the prescribed methods, techniques and support tools
General knowledge	• Higher vocational education or academic work and thought level, study related to ICT • ICT knowledge: – familiar with development methods (SDM, DSDM, RAD, CBD) – knowledge of system design techniques (functional design and technical design) – knowledge of ITIL processes – experience with project management methods; – experience with word processors and spreadsheets – experience with test support tools

Technical knowledge	• Knowledge of the test methods to be used • Knowledge of basis techniques of structured testing • Practical experience in the use of test analysis techniques • Practical experience in the evaluation of test specifications • Practical experience in the use of checklists • Practical experience in the use of inspection techniques • Practical experience in the detection of anomalies in the documentation and systems • Practical experience in the use of issue administrations • Knowledge of concepts for automated testing • Knowledge of the use of automated test tools
Skills	• Creative • Accurate • Independent • Planning and organisation • Problem analysis and judgement • Team worker • Can handle stress • Good communication skills
Work experience	• Minimum of 1 year's experience as a tester in the ICT sector is required

TESTEXECUTER	
Function description	The test executor is responsible for the execution of the test using the test scenario and test design produced by the test analyst. The test executor also assists with reviewing the design documents. He registers all findings that relate to the deviations found between the information system and/or the system design and/or the test design. Next to that he communicates and reports to the people involved. In his job the provided methods, techniques and tools support the tester
Tasks, responsibilities and authorities	• Intake of the system components to be tested: short inspection concerning the completeness of the system components to be tested • Intake of the test scenario and test cases: short inspection concerning practicability of the test cases • Preparation of test execution: fill initial state files in conformity with specifications • Execution of the tests: – dynamic testing based on the test cases – static testing, like reviews based on checklists • Record results • Review the results

	• Administer the findings • Report to test management • Manage testware: daily collection, selection and optimising of testware • Using prescribed methods, techniques and support tools
General knowledge	• Higher or intermediate vocational education work and intelligence level • ICT knowledge: – knowledge of development methods (SDM, DSDM, RAD, CBD) – knowledge of ITIL processes – experienced with project management methods – experienced with word processors and spreadsheets
Technical knowledge	• Knowledge of the prescribed test methods • Practical experience with reviewing test specification • Practical experience with checklists • Knowledge of inspection techniques • Practical experience with the detection of anomalies in documentation and systems • Practical experience with the use of a findings procedure

Other skills	• Creative • Accurate • Independent • Planning and organisation • Problem analysis and judgement forming • Team player • Can handle stress • Good communication skills
Work experience	• Minimum of 1 year’s work experience in ICT desirable

TEST NAVIGATOR

Function description

The test navigator is responsible for navigation through the test scripts that support the automated test execution. With support from the test support tools and the use of macros, manual actions (keystrokes and mouse movements) performed by the test executor are automated. For this, the test navigator records the manual actions in a navigation script that functions as a controlling program

Tasks, responsibilities and authorities

- Intake:
  - test cases
  - test objects
- Test support tools: responsible for the installation of test support tools and other supporting items
- Test scripts:
  - specification of test scripts
  - programming of the test scripts
  - inspection and testing of the test scripts
- Initials state files: making scripts for the creation of test data based on specifications
- Testing:
  - drawing up test sets with test scripts
  - execution and support of automated tests

	• Administration of findings • Testware related to automated testing: – collation and composition of the testware – selection of testware for maintenance – transfer of testware to the test analyst
General knowledge	• Higher vocational education or academic work and thought level, study related to ICT • ICT - knowledge – knowledge of development methods (SDM, DSDM, RAD, CBD) – knowledge of structured programming concepts – knowledge of programming languages – knowledge of ITIL processes – experienced with word processors and spreadsheets.
Technical knowledge	• Knowledge of the prescribed test methods • Knowledge of the test analysis techniques • Knowledge of basic structured testing techniques • Practical experience with the use of support tools

	• Practical experience with checklists • Practical experience with the detection of anomalies in documentation and systems • Practical experience with the use of findings procedures • Knowledge of automated testing concepts • Proven ability in using automated test support tools
Other skills	• Creative • Flexible • Accurate • Results driven • Problem analysis • Team player • Can handle stress • Persistent • Good communication skills.
Work experience	• Minimum of 1 year's work experience in ICT desirable

TEST CONSULTANT

Function description	The test consultant advises and supports the optimal practice of test methods, test techniques and test support tools to project members (in test role or not). In this way he helps set up test projects and outlines the test strategy. He enthuses clients and project members with the goal to improve the test approach within the projects
Tasks, responsibilities and authorities	• Advise on test phasing • Advise on test organisation • Advise on test infrastructure (test environments) • Advise on and support for applying test methods, test techniques and test support tools • Advise on applying automated testing • Advise on reuse of testware • Advise on various inspection techniques • Testing of and feedback on the usability of test methods, test techniques and test support tools • Stimulate collecting, analysing and distributing of data (metrics) in projects • Support test leaders with setting up the test plan and test (final) report.

General knowledge	• Higher vocational education or academic work and thought level, study related to ICT • ICT - knowledge – knowledge of development methods (SDM, DSDM, RAD, CBD) – knowledge of system design techniques (functional design and technical design) – knowledge of ITIL processes – experienced with project management methods – knowledge of test support tools – knowledge of quality assurance – knowledge of risk management – knowledge of prototyping.
Technical knowledge	• Thorough knowledge of and experience with test methods, test techniques and test support tools • Ability to combine (test) theory and pragmatism • Very thorough knowledge of, experience with and ability to use test methods, test techniques and test support tools • Practical experience with applying test process improvement models (TMM, TPI) • Practical experience with moderating group inspections (Fagan inspections)

	• Practical experience with performing audits on test plans • Practical experience with applying concepts for automated testing • Practical experience with use of test navigation support tools • Practical experience with advising on subjects mentioned above
Other skills	• Creative • Daring and self-confident • Customer focused • Listening and sensitivity • Verbal skill of expression and presentation • Writing skills • Awareness of surroundings (expert knowledge) • Organisational sensitivity • Force of conviction • Independent • Pragmatic strain
Work experience	• Significant experience as test analyst (minimum 4 years) or test manager (2 years) • Experience in a leading role is desirable

J

Template for Progress Report

This template describes the subjects that a test manager should include in progress reporting. He writes the report on a regular basis during the execution of a test level.

A progress report should be brief and the subjects covered should be immediately clear to the client and stakeholders. These documents do not have a formal layout with version and configuration management and contents list.

General Progress

Provide a brief general summary of progress here. Remember also to provide a brief summary on the outstanding issues, connected to product risks.

Activities/Milestones/Products in This Period

Provide an overview here of activities carried out and products delivered. Information on completed activities can be obtained from, for example, the overview from the earned value method. Base this on the latest version of the planning. The following categories could be used:

Planned/Completed
Provide a brief summary of delivered, planned products and executed, planned activities. For example:

- test analysis of cluster XYZ functionality completed;
- preparation of cluster ABC efficiency completed;
- test environment set up by support department.

Planned/Not Completed
Provide a brief summary of planned, but not delivered, products and planned, but not executed, activities *including reasons why not.* For example:

- Preparation of test cluster KLM reliability not started owing to illness of test analyst.

Not Planned/Completed
It could happen that within a test project unplanned activities have to be carried out to "start up" the test project or to keep it going. This could well be activities that are outside the scope of the test project but are necessary for the progress of the project. Mention these activities and how much time they take. Also mention the consequences for the planning.

Activities/Milestones/Products of the Coming Period

Provide an overview of the products to be delivered and activities to be carried out according to planning. If it is already known that certain parts of the plan cannot be realised, it should be reported here. For example:

- Review of cluster XYZ functionality;
- test cluster of KLM reliability, to be delivered, may be delayed owing to illness of test analyst;
- intake test of test environment.

Quality of the Information System

An overview should be given here of the status of the quality of the information system. This may be done by including a separate table as shown in the example below (note that only those product risks are included in the table for which the requirements are to be tested at this test level).

The status may also be recorded in a separate document, or a link may be made between the test case documentation and the product risks, and the status documented here with reference made to this supplementary documentation. At any rate, provide a brief summary here. For example, none of the product risks have been covered as yet as the actual test execution has not yet started. The quality of the information system does not meet the agreed acceptance criteria.

Product risk	**MoSCoW**	**Requirement**	**Test conditions**	**Status**
Customer cannot perform a transaction	Must test	Customer can perform transaction via own bank	XYZC1, 2	Open
		Customer can perform transaction via host bank	XYZC3, 4	Closed
		Customer can select from standard amounts	XYZ5–8	Reopen
		Customer can select amount/denomination of notes	XYZ 9–14	Open
...	...	...	...	...

The stakeholder wishes to know when an important part of the functionality or goals of the new system will be available. If you have insight into this, report on it!

How do you obtain the insight? As soon as product risks 1, 2 and 3 have tested successfully, functionality A is available for release. This link between product risks and the stakeholder's required functionality or goals can be seen from the cluster cards if the cluster has been arranged by system parts. The product risks are therefore noted in the cluster card for the particular system part (note that this can only apply to some of the quality attributes).

This insight also provides the project manager with a management tool: if another product risk has to be tested to realise a stakeholder's goal, priority can be given to it here during execution.

If you have met the entry or exit criteria set for the test level during this period, you can report that here.

Budget and Progress

Provide a brief summary here of the budget already spent and still to be spent. Also use the forecasting capacity of the earned value method (EVM) to indicate whether the budget is still adequate. Add the graphic overview of progress and budget spending from EVM (this can be found in the test control matrix).

Risk Management (Project Risks)

Provide an overview here of the project risks documented in the test plan and their status (as in the test plan, report here only the 5 to 10 most important project risks). When a project risk has been dealt with, it remains in one progress report with status closed. Thereafter, it will disappear from the list. New risks that arise in the course of the project will of course be added to this list.

Project risk	**Importance**	**Measures**	**Responsibility**	**Date**	**Status**
Tester Jack will leave the project in 2 weeks. Replacement still to be found who fits the profile	High	Tester Gary will take over tasks a and d. Recruit new tester with slightly adjusted profile who will take over the remaining tasks and who can support Gary	Resource management	A new employee with adjusted profile must at any rate be present for transfer in 1 week 's time (31 Jan. 03)	Open
Connection between Unix and mainframe platform in test environment not present. Management confronted with production faults, which took priority	High	Free up an employee in consultation with management, so that he may provide full-time support to all the test projects and cannot be claimed for production	Head of Management	29 Jan. 03	Closed

Points of Focus and Problems

Situations sometimes require action at short notice. They have then to be put before the parties concerned (this is done in a separate exception report). A summary for the client can then be included in the progress report. This may be a project risk that has been foreseen and has now occurred. A situation may also arise that has not been foreseen and for which no risk countermeasures have been devised. Reflect the following points:

- description of the situation or the problem;
- description of the cause of the situation;
- the consequences for the test project as regards time, money or quality;
- a summing up of the alternatives for dealing with the situation and for each alternative the influence it will have on time, money, quality and any extra project risks that will arise as a result;
- a recommendation for an alternative, with supporting information;
- decision: if no decision has yet been taken, this will of course not yet be completed, but reference should be made to the following report which will contain the decision.

K

Template for Phase Report

This template describes the subjects that a test manager should include in a phase report. The test manager draws up a phase report following conclusion of a test level. If a test project consists of only one test level, the phase report and final report are one and the same.

This template does not cover general topics such as version management and configuration management.

A brief description is provided of the content of each subject.

K.1 Management Summary

Provide a brief summary of the report. Be sure to report on:

- Whether the goals of this test level as stated in the test plan have been achieved.
- The quality of the information system based on risk- and requirement-based testing.
- An overview and analysis of the differences between planned and realised in relation to the following aspects:
  - Time
  - Budget
- Recommendation for proceeding to the following test level.

K.2 Test Assignment Test Level

Describe here the original assignment for this test level. This will be set against the actual realisation in the recommendations.

K.3 Quality of the Information System

This section provides an overview of the quality of the information system. Based on this overview, a recommendation is made in Sect. 5 concerning proceeding to a subsequent test level.

K.3.1 Product Risks and Requirements

Below is an overview of the product risks and requirements that have been covered and those that have not been covered by the test execution. This information is obtained from the progress report of the test level. With the product risks that do not have the status "correct", indicate why this is so. If it is due to an outstanding issue, this will be explained in the next section. It is also possible that a product risk cannot be tested owing to the materialisation of a project risk.

Product risk	**MoSCoW**	**Requirement**	**Test conditions**	**Status**
Customer cannot perform a transaction	Must test	Customer can perform transaction via own bank	XYZC1, 2	Correct
		Customer can perform transaction via host bank	XYZC3, 4	Correct
		Customer can select from standard amounts	XYZ5-8	Correct
		Customer can select amount/denomination	XYZ 9-14	Correct
…	…	…	…	…

K.3.2 Brief Overview of Issues

Provide a brief summary of the issues still outstanding, including the related product risks and the importance of these (issues are linked to test conditions; product risk and importance can be seen from the table in Section K.3.1). Refer to the issue administration for a complete overview of the issues, or include these as an appendix.

K.3.3 Entry and Exit Criteria

Provide here an overview of all the entry and exit criteria for the test level and indicate to what extent these have been met.

Mention the entry criteria for the test level explicitly, since failure to (fully) meet an entry criterion (e.g. the delivery by the support department of a test environment according to specifications) may delay the project or result in an inability to test a product risk during the test level.

An exit criterion of this test level is often an entry criterion for the following test level. Therefore, provide an explanation when exit criteria have not been met and state what the consequences of this are.

K.4 Planned Versus Realised

K.4.1 Time

Indicate (based on the data from the earned value method (EVM) and the test control matrix (TCM) or the time registration of the project) which discrepancies exist between planned and realised time per part activity. If there is familiarity within the organisation with the TCM method, this can be included here. For convenience it is often necessary to arrange these details in a table (as shown below). The part activities may, for example, be obtained from MS Project.

Activities	Planned hours	Actual	Over/under	Remarks
Test strategy				
Test plan				
Test environment				
Test design				
Test execution				
Test management				
Test automation				
Total				

K.4.2 Money

Provide an overview of the budget and actual spending. For this use an overview of the planned activities and the budgets planned for these. The level of detail in this table depends of course on the level of detail to which the budget is divided across the various activities. Information on these figures can be gleaned from EVM and TCM (or from the above table by calculating hours × cost).

Activities	Budget	Actual	Over/under	Remarks
Test strategy				
Test plan				
Test environment				
Test design				
Test execution				
Test management				
Test automation				
Total				

K.4.3 Deliverables

Provide a summary here of the products that were delivered by this test type. The following test type will use a number of the products again. Other products will be transferred to maintenance after completion of the total test project (the deliverables of the total test project are noted in the test plan).

An overview of the delivered products can be obtained from the progress reports.

K.5 Recommendations

In this section provide a recommendation on proceeding to a subsequent test level. It is of course necessary to support your recommendation (base your recommendation on Section K.3).

Compare the original test assignment to what has actually been realised in the test level. Describe and support the discrepancies (if everything has not been completed in this test level, it does not mean by definition that it is not possible to proceed to the following test level).

Particular actions that still have to be completed before going on to a subsequent phase can also be mentioned here. A recommendation on the current outstanding issues and project risks, and the treatment of these in the following phase, also belong in this section.

L

Template for Final Report

This template describes the subjects that a test manager should include in a final report, which is written up after the conclusion of a test project.

This template does not cover general topics such a version management and configuration management.

A brief description is provided of the content of each subject.

L.1 Management summary

Provide a brief summary here of the report. Outline the following points:

- Whether the goals of the test projects as a whole, as stated in the project test plan, have been met (refer here to the test assignment).
- The quality of the information system on the basis of Risk & Requirement Based Testing.
- Overview and analysis of the discrepancies between what was planned and what was realised, in relation to the following aspects:
  - time;
  - budget.
- The most important realized benefits of the project.
- Recommendation for transferring the information system into production (Go/NoGo decision).

L.2 Test assignment

Describe the original test assignment here from the project test plan. These will be compared against the actual realisation in the recommendations.

L.3 Quality of the information system

This section provides an overview of the quality of the information system. Based on this overview, a recommendation can be given in Section L.5 on putting the information system into production.

L.3.1 Product risks and requirements

An overview is provided below of the product risks and requirements that have been covered and those that have not been covered by the test execution. This information is obtained from the progress report or phase reports on the various test levels. Provide an explanation next to the product risks that do not have the status "correct" for why this is so. If it is due to an outstanding issue, this is explained in the following section. It is also possible that a product risk could not be tested because of the materialisation (and apparently delayed solution) of a project risk. More so than in a phase report, it is important to explain here why certain product risks do not have the status "correct", and what the possible consequences of this are.

Product risk	**MoSCoW**	**Requirement**	**Test conditions**	**Status**
Customer cannot perform transaction	Must test	Customer can perform transaction via own bank	XYZC1, 2	correct
		Customer can perform transaction via host bank	XYZC3, 4	correct
		Customer can select from standard amounts	XYZ5-8	correct
		Customer can select amount/denomination of notes	XYZ 9-14	correct
…	…	…	…	…

L.3.2 Overview of issues

Provide a brief summary of the outstanding issues, including the related product risks and their importance (issues are linked to test conditions, and product risk and importance can be seen from the table above).

Also indicate what is being done with the outstanding issues. Solving them within a planned subsequent test level is not an option.

Are solving and testing these being moved to a subsequent release of the system, or will they be covered in regular maintenance? Who will be responsible for monitoring this? Organising the transfer of these issues is mandatory!

Will the issues be closed?

Required actions issuing from this that fall outside the scope of the current test project should be reported in the section on recommendations.

Optionally, provide an overview of the number of issues according to test level (this can be used in an evaluation report for analysis, e.g. to compare the number of issues against the costs of the test level).

Refer to the issue administration for a complete overview of the issues, or include them as an appendix.

L.3.3 Acceptance criteria

The acceptance criteria are documented in the test strategy (and subsequently translated in the test plan into entry and exit criteria for the various test levels). Provide an overview here of all the acceptance criteria. Indicate in this whether or not they have been met, with an explanation of reasons and consequences. If possible, link them to the product risks.

L.4 Costs and benefits of the test project

L.4.1 Costs in time

In the phase reports on the test levels, an overview is given of the costs in time and money relating to the test level. In this document, provide an overview of the costs of the whole test project. Include these data and add them up below.

Another method, similar to the phase reports, is as follows: indicate (based on information from the earned value method (EVM) and the test control matrix (TCM) or the time registration of the project) what discrepancies there are between planned and realised time per activity. If the organisation is familiar with the TCM method, this can be included here. For convenience, it is often necessary to arrange these details in a table (as shown below). The activities, for example, can also be obtained from MS Project.

Activities	Planned hours	Actual	Over/under	Remarks
Test strategy				
Test plan				
Test environment				
Test design				
Test execution				
Test management				
Test automation				
Total				

L.4.2 Costs in money

Provide an overview of the budget and actual spending. For this use an overview of the planned activities and the budgets planned for these. The level of detail in this table will of course depend on the level of detail at which the budget is divided across the various activities. Information for these figures can be obtained from EVM and the TCM (or from the above table by calculating hours × cost).

Do not forget the overall costs, such as the setting up of communal test environments, project audits, travel costs, meeting costs, costs for tooling and consultancy, etc.

Activities	Budget	Actual	Over/under	Remarks
Test strategy				
Test plan				
Test environment				
Test design				
Test execution				
Test management				
Test automation				
Total				
Other costs				
Travel expenses				
Tool for issue management				
…				
Total				

L.4.3 Benefits of the test project

It is difficult to reflect the benefits of a test project, to say nothing of quantifying them.

One possibility is to make a link between an issue and a product risk, via a test condition. If the fault in the information system had not been found and solved, would the product risk have occurred in the production environment? What would it have cost the organisation to solve this in production? How much time and effort went into solving the issue? What are the costs of (part of) the system being out of use? Would the fault in the system damage the company's image? Would the customers go elsewhere?

Deliverables

Provide an overview of the deliverables of the test project. Discrepancies in relation to the test plan should be explained: why has a deliverable that is stated in the test plan not been delivered, and vice versa?

Costs and benefits of test automation

The costs of test automation are shown in Sects. L.4.1 and L.4.2 above.

In order to determine the benefits of the test automation, compare it with the costs of manual test execution. This can be determined by looking at how much the manual execution of comparable clusters/test conditions has cost (also look at how often a particular test has been carried out: with repeat testing, test automation often pays).

L.5 Recommendations

In this section, provide a recommendation on whether or not to transfer the information system to production (Go/NoGo Decision).

Compare the original test assignment as described in Section L.2 with what has been realised (in the various test levels collectively).

Describe and support the discrepancies.

Formulate the recommendations. It is of course necessary to support them (base the recommendations on Section L.3). Remember, too, to provide a recommendation on the treatment of the outstanding issues and project risks.

Glossary of Terms

Acceptance criteria

Criteria that stakeholders define at the start of a test project. They use these criteria at the end of a test project to accept the information system.

Basic test

The first step in test execution. The testers try to execute every test. But a test might cause a (blocking) issue. If issues occur, multiple retests are necessary.

Black-box test

A test based on the analysis of the specifications of an information system, without having knowledge of the internal structure of this system.

Business controls

The variables the test manager possesses to control the test project. These are: time, money and quality. The project manager possesses an extra control, namely the size of an information system.

Business impact

The consequences of an issue for daily practice.

Chain test

A test in which a whole chain of systems from the first input to the last output is regarded. Synonym: end-to-end test.

Cluster

Logical test unit.

Cluster card

A card containing essential information (i.e. stakeholder, product risks and acceptance criteria) necessary for designing and executing tests related to a cluster. Together with creating the cluster matrix, creating the cluster cards is the last step in describing a test strategy. It is the basis for test analysis.

Cluster matrix

First assignment of quality attributes to be tested to stakeholder and test level. This division indicates which cluster cards will be made.

Complete test

Within the complete test the testers can execute all tests. No issues with test impact remain.

Detailed test plan

A test plan per test level. This test plan is an elaboration of the project test plan when a test project consists of more than one test level.

Dynamic testing

Testing by processing actions on the information system to be tested.

Earned Value Method (EVM)

Method for measuring the progress of a (test) project. The starting point is that progress is measured not only on the basis of hours spent compared with hours estimated (money), but also in terms of actual progress: the timely delivery of products (time). Therefore, budget spend and progress are separated. One characteristic is that a product is only booked as realised when 100% complete.

Entry criteria

Criteria that a test manager defines to start a specific test level. Every test level can have several entry criteria. These entry criteria are described in the project test plan or the detailed test plans.

Error guessing

A test technique (usually unstructured) with which faults in the information system are looked for, based on experience with the system.

Estimation

The definition of the boundaries in time and money within which the testing is allowed to operate.

Evolutionary planning

In this method the total project is divided into smaller parts. Only the part in the near future is planned for in detail; the other parts are only globally planned. After completing a part, the next part is planned in detail.

Exception report

A formal report from the test manager to the client stating a situation that needs to be solved at short notice. For example, a project risk that materialised. Also a situation might occur that was not foreseen and for which no countermeasures were defined.

Exit critera

Criteria the test manager defines to end a specific test level. Every test level can have several exit criteria. These exit criteria are described in the project test plan or the detailed test plans.

Final report

The report the test manager makes after completion of all test levels. The final report contains advice to the stakeholders. In this report the test manager gives an overview of the results of the different test levels. In addition, the extent to which the acceptance criteria are met is an important part of this final report.

Final test

Last in the series of retest runs. In this last retest run the testers check whether all issues that should be solved are solved.

Functional acceptance test

Testing the documented and implied functions (does the system do what it should do?). This test level is based on the functional specifications.

Goal–question–metric (GQM) method

A founded metrics program.

Incident management

See issue management.

Intake test

A test that is performed before the actual test execution. The test team checks whether the information system meets the stated quality (*see* entry criteria). Successfully completing the intake test means that the actual testings can start. If the test is not completed successfully, the system is sent back to the developers.

Integration test

Testing the communication between programs (integration test “in the small”) or complete information systems (integration test “in the large”).

Issue

Deviation between expected and actual outcome of a test.

Issue management

Managing the issues that occur during test execution via procedures and tooling. It is also known as incident management.

ITIL (Information Technology Infrastructure Library)

A widely used standard in practice for setting up a maintenance process for information systems.

Joint Testware Development (JTD)

A method the test manager and testers can use to develop testware in case good documentation is missing. This information is collected during a brainstorm session following specific rules.

Logical Unit of Test (LUT)

A uniform collection of interrelated activities from test design to test execution.

Metrics

A metric is a quantified measure of a process or product attribute that is characteristic of the product or process object to be measured.

MoSCoW priority

Classification of test priorities based on product risks: Must test, Should test, Could test and Won’t test.

Phase report

A formal report from the test manager to the client after completion of a predefined phase of the test project. This might be the completion of a test level.

PRINCE2

PRINCE2 is a project management method that is generally applicable to projects. The name PRINCE stands for PRojects IN Controlled Environments.

Production acceptance test

Tests whether the system can be exploited.

Product risk

A risk that is directly related to an information system. This risk can be covered through testing.

Progress report

The report the test manager draws up during the execution of the test project.

Project risk

A risk that relates to the set-up and control of a (test) project.

Project test plan

The overall test plan with a description of all test levels in a test project. The various test levels are elaborated in detailed test plans.

Regression test

A test to check whether all important unchanged parts of the information system still function as before following a change in the information system.

Requirement-based testing

Testing based on the requirements (functional and non-functional) of the information system.

Risk- & Requirement Based Testing (RRBT)

Test approach in which the product risks and requirements are combined in a structured way. This combination is the basis for setting up and controlling a test project.

Quality attribute

Describes the wishes and demands for an information system. Six attributes with various subattributes are described in ISO9126.

Quick scan

The test manager follows the Test Management Model for the first time in a short period of time. This gives him a first impression of the already existing testware, procedures and tools.

Stakeholder

An employee or department that has a direct interest in the correct operation of an information system. Synonym: interested parties.

Static testing

Executing tests without running the information system. For example, reviews or inspections of specifications or code.

Strategic test slicing method (STSM)

A method to slice tests according to predefined criteria in the case of shortage in time/money. This relates to the priority of the product risks.

System test

Test by the system developer in a laboratory environment to demonstrate whether the system (or parts of it) is developed according to the functional and technical specifications.

Test case

All pre- and post-conditions, test data and expected outcomes of a test. A test case is designed to verify a specific test condition.

Test centre

The test centre is a part of the organisation that designs and executes the tests. The test centre can also play a supervising role by assigning the test process a certain approval or certifying it before a system is taken into production. The test centre is a quality department with regard to testing.

Test competence centre (TCC)

The TCC is a staff department. The TCC centralises several test services, e.g. test methodology, resources, test advice, test automation, education and support.

Test condition

A level under clusters. It indicates what will be tested for a cluster. Test conditions are related to requirements with their accompanying product risks.

Test control matrix (TCM)

Helps to keep account of and presents the business controls of time, money and quality and the dependencies between the test project and development.

Test effort estimation model (TEEM)
A model for estimating test projects based on metrics.

TestFrame
LogicaCMG's approach to structured testing.

Test impact
Indicates the consequences of an issue for the execution of the test. The division is: stopper, test obstructing and non-obstructive.

Test level
A division of the total testing necessary at various levels. These levels are related to stages in system development and those executing the tests, e.g. unit test or functional acceptance test.

Test management file
The file in which the test manager stores all information that is important to the test project. During the term of the test project, all project members can consult this file for the latest updates. After closing the test project, the file will be transferred to a maintenance department.

Test Management Model
LogicaCMG's test management model used as a basis for this test management book. It describes the activities of the test manager from preparation to execution and closure of the test project.

Test run
The period in which a related group of test clusters is executed for a version or build of the information system.

Test strategy
Describes the scope and depth of the test: which product risks will be covered within the test project and which not.

Test technique
A collection of actions to derive test conditions and/or test cases in a reproducible and objective way.

Test type
Test activities aimed at testing an information system on a specific (combination of) quality attribute(s).

Testware
This term contains all products that are developed within a test project.

User acceptance test
Testing whether the end users can work with the system. This test is based on the end users' procedures.

White-box test
Test based on the internal structure of an information system.

Bibliography

Books

Beizer, Boris: *Black-Box Testing – Techniques for Functional Testing of Software and Systems* (Wiley, New York, 1995)

Bersoff, Edward H.: *Elements of Software Configuration Management* (IEEE Computer Society Press, Silver Spring, MD, 1984)

Black, Rex: *Managing the Testing Process* (Microsoft Press International, Redmond, WA 1999)

Boehm, Barry W.: *Software Engineering Economics* (Prentice Hall, Upper Saddle River, NJ, 1981)

Buwalda, Hans, Dennis Janssen & Iris Pinkster: *Integrated Test Design and Automation Using the TestFrame Method* (Addison-Wesley, Harlow, 2002)

CCTA (Central Computer and Telecommunication Agency): *Best Practice for Service Support – ITIL the Key to Managing IT Service* (The Stationery Office, London, 2000)

CCTA (Central Computer and Telecommunication Agency): *Managing Successful Projects with Prince 2* (The Stationery Office, London, 2002)

CMG: *TestFrame: Een praktische handleiding bij het testen van informatiesystemen* (ten Hagen & Stam, Den Haag, 2001)

Collard, Ross: *Realistically Estimating Software Testing Projects* (Collard, New York, 2003)

Coul, Johan C. op de: *Beheer van de informatievoorziening – Een bedrijfskundige benadering* (Samsom Bedrijfsinformatie, Alphen aan den Rijn, 1996)

Craig, Sue: *Make Your Mark, Influencing Across the Organization* (McGraw-Hill, Maidenhead, 1997)

Daich, G., G. Price, B. Ragland & M. Dawood: *Software Test Technologies Report* (Software Technology Support Center, Utah, August 1994)

DeMarco, Tom & Timothy Lister: *Peopleware – Productive Projects and Teams*, 2nd edn (Dorset House, New York, 1999)

Deming, W. Edwards: *Out of the Crisis* (MIT Press, Cambridge, MA, 2000)

Dustin, Elfriede, Jeff Raska & John Paul: *Automated Software Testing: Introduction, Management, and Performance* (Addison-Wesley, Boston, MA, 1999)

Fewster, Mark & Dorothy Graham: *Software Test Automation: Effective Use of Test Execution Tools* (Addison-Wesley, Boston, MA, 1999)

Gerrard, Paul & Neil Thompson: *Risk-Based E-Business Testing* (Artech House, Norwood, MA, 2002)

Gilb, Tom: *Principles of Software Engineering Management* (Addison-Wesley, Harlow, 1988)

Gilb, Tom & Dorothy Graham: *Software Inspections* (Addison-Wesley, Boston, MA, 1993)

Hatton, Les: *Safer C: Developing Software for High-Integrity and Safety-Critical Systems* (McGraw-Hill, Maidenhead, 1995)

Helmer, O.: *Social Technology* (Basic Books, New York, 1966)

Hetzel, Bill: *The Complete Guide to Software Testing*, 2nd edn (Wiley, New York, 1988)

IFPUG (International Function Point User Group): *Function Point Counting Practices* (release 4.1, January 1999)

ISO/IEC 9126-1: *Software engineering – Software product quality – Part 1: Quality Model* (International Organization for Standardization, Geneva, 2001)

Institute of Electrical and Electronics Engineers: *IEEE Standard Classification for Software Anomalies* (IEEE Std 1044 1993)

Institute of Electrical and Electronics Engineers: *IEEE Standard for Software Test Documentation* (IEEE Std 829 1998)

Kaner, Cem, James Bach & Bret Pettichord: *Lessons Learned in Software Testing: A Context-Driven Approach* (Wiley, New York, 2002)

Kaner, Cem, Jack Falk & Hung Quoc Hguyen: *Testing Computer Software*, 2nd edn (Wiley, New York, 1999)

Kit, Edward: *Software Testing in the Real World: Improving the Process* (ACM Press, New York, 1995)

Koomen, Tim & Martin Pol: *Test Process Improvement, leidraad voor stapsgewijs beter testen* (Kluwer Bedrijfsinformatie, Deventer, 1998)

Koppens, Sander & Bas Meyberg: *Operationeel beheer van informatiesystemen* (Kluwer Bedrijfsinformatie, Deventer, 2000)

Kor, Rudy: *Werken aan projecten – Een handreiking*, 3rd edn (Kluwer Bedrijfsinformatie, Deventer, 1999)

Marik, Brian: *The Craft of Software Testing – Subsystem Testing, Including Object-Based and Object-Oriented Testing* (Prentice Hall, Englewood Cliffs, NJ, 1995)

Martens, Marlies, Kees-Jan Groen & Bertram van der Wal: *Publiek Management: 65 modellen* (FC Klap, Utrecht, 2002)

Mors, N.P.M.: *Kwaliteitszorg door acceptatietesten* (Lansa, Leidschendam, 1994)

Myers, J. Glenford: *The Art of Software Testing* (Wiley, New York, 1979)

Onna, M. van, B. Hendriks & G. Schraven: *De kleine PRINCE 2, projectmanagement methodiek voor kleine en middelgrote projecten*, 2nd edn (ten Hagen Stam, Den Haag, 2000)

Patton, Ron: *Software Testing* (Sams Publishing, Indianapolis, 2000)

Perry, William E. & Randall W. Rice: *Surviving the Top Ten Challenges of Software Testing: A People Oriented Approach* (Dorset House, New York, 1997)

Paulk, Mark C., Charles V. Weber, Bill Curtis & Mary Beth Chrissis: *The Capability MAturity Model – Guidelines for Improving the Software Process* (Addison-Wesley Longman Inc., 1994)

Pol, Martin, Ruud Teunissen & Erik van Veenendaal: *Testen volgens Tmap*, 2nd edn (Uitgeverij Tutein Nolthenius, Den Bosch, 2000)

Solingen, Rini van & Egon Berghout: *The Goal/Question/Metric Method* (McGraw-Hill, Maidenhead, 1999)

Stapleton, Jennifer: *DSDM: The Method in Practice* (Addison-Wesley, London, 1997)

Veenendaal, Erik van: *The Testing Practitioner* (Uitgeverij Tutein Nolthenius, Den Bosch, 2002)

Watkins, John: *Testing IT: An Off-the-Shelf Software Testing Process Handbook* (Cambridge University Press, Cambridge, 2001)

Wijnen, Gert & Rudy Kor: *Het managen van unieke opgaven – samen werken aan projecten en programma's* (Kluwer Bedrijfsinformatie, Deventer, 1997)

Articles

Amland, Stale: *Risk Based Testing and Metrics* (5th International Conference EuroSTAR '99, 8–12 November 1999, Barcelona, Spain)

Bach, James: *Heuristic Risk-Based Testing* (Software Testing & Quality Engineering, November 1999)

Black, Rex: *Effective Test Status Reporting – The Upward and Outward Management of Communicating Test Progress and Results* (Software Testing & Quality Engineering, March/April 2000, Vol. 2, Issue 2)

Burnstein, Ilene, Taratip Suwannasart & C.R. Carlson: *Developing a Testing Maturity Model: Part I* (Crosstalk, STSC, Hill Air Force Base, Utah, August 1996, pp. 21–24)

Burnstein, Ilene, Taratip Suwannasart & C.R. Carlson: *Developing a Testing Maturity Model, Part II* (Illinois Institute of Technology)

Chavali, Sreeram Kishore: *Approach to Implementing Risk Based Testing* (19 March 2001)

Collard, Ross: *Speeding the Software Delivery Process, Part 1, Manage and Strengthen Testing* (stickyminds.com)

Collard, Ross: *Speeding the Software Delivery Process, Part 2, Conduct Early and Streamlined Testing* (stickyminds.com)

Collard, Ross: *Speeding the Software Delivery Process, Part 3, Manage the Risks and the Process* (stickyminds.com)

Derby, Esther: *Risky Beginnings* (Software Testing & Quality Engineering, November/December 2000)

Gerrard, Paul: *Risk-Based E-Business Testing, Part 1: Risks and Test Strategy* (2000)

Gerrard, Paul: *Risk: The New Language of E-Business Testing* (Systeme Evolutif Ltd., version 1.0, STAREAST 2001)

Gerrard, Paul: *Risk-Based Test Reporting* (14 February 2002)

Gilb, Tom: *Risk Management: A Practical Toolkit for Identifying, Analyzing and Coping with Project Risks*, version 3.1 (Quality Week, San Jose CA, 1999)

Goodwin, Steve: *Software Risk Management Makes Good Business Sense* (Software Release Magazine, Q4, 2000)

Horne, Geoff (Integrity Software Testing & Quality (NZ) Ltd.): *Testing in a Squeezed, Squeezed World* (February 2002)

Janssen, Dennis: *Developing Software Together, Using the JTD Method,* (Test2000 Conference, 2000)

Janssen, Dennis: *Kwaliteit van testen: onbeheersbaar of ongecontroleerd* (Software Release Magazine, No. 8, December 2002, pp. 32–37)

Kandler, Jim: *Product Risk Analysis Clarifies Requirements* (STAREAST 1999)

Kroll, Per & Philippe Kruchten: *From Waterfall to Iterative Development – A Challenging Transition for Project Managers* (SM/ASM 2002)

Potter, Neil & Mary Sakry (The Process Group): *Planning for Project Surprises – Coping with Risk* (Pacific Northwest Software Quality Conference, 1999)

Rothman, Johanna: *Successful Test Management: 9 Lessons Learned,* (STARWEST 2000)

Rothman, Johanna (Rothman Consulting Group, Inc.): *What to Do When the Right Person Doesn't Come Along* (Conference, SM/ASM 2000)

Slade, Randy (Kaiser Permanente Information Technology): *How To Break out of the Same Old Routine of Bad Quality* (STARWEST 2001)

Statz, Joyce & Susan Tennison: *Getting Started with Software Risk Management* (American Programmer, Vol. 8, No. 3, March 1995)

Veenendaal, Erik van: *Testing Maturity Model: Van detectie naar preventie* (Software Release Magazine, Vol. 6, No. 8, December 2001)

Others

Bach, James: *How Much Testing Is Enough?* (stickyminds.com, 17 September 2002)

Graham, Dorothy: *Test Is a Four-Letter Word* (TestNet presentation, 1999)

Managing Successful Projects Prince 2 Electronic Manual (Central Computers and Telecom Agency & Key Skills Ltd., 1999)

Smith-Brock, Jennifer: *Using Risk Analysis to Prioritize Your Testing* (EuroSTAR 2001)

Test Strategy Questionnaire (ISEB Practitioner, Improve Quality Services)

US Air Force Systems Command: *Cost-Schedule Management of Non-major Contracts* (AFSCP 17303, Andrews AFB, Maryland, November 1987)

Index