TMap® Next

for result-driven testing

Tim Koomen, Leo van der Aalst, Bart Broekman, Michiel Vroon

UTN Publishers

SOGETI TMAP

TMap® Next

TMap® Next

for result-driven testing

Tim Koomen
Leo van der Aalst
Bart Broekman
Michiel Vroon

Uitgeverij Tutein Nolthenius

UTN Publishers, 's-Hertogenbosch

Second run 2007

This book is a full translation of the the Dutch book
TMap Next, voor resultaatgericht testen, 2006, ISBN 90-72194-79-9

Translation: V V H business translations, Utrecht (NL)

UTN Publishers
Willem van Oranjelaan 5
5211 CN 's-Hertogenbosch
The Netherlands
www.utn.nl

ISBN(10) 90-72194-80-2
ISBN(13) 9789072194800

Preface

Foreword to TMap Next, by Rex Black

By the 1980s, Fred Brooks and Barry Boehm had explained that the costs and risks associated with bugs are high and increase as projects progress. In the 1980s, Boris Beizer and Bill Hetzel added another essential insight: testing, done properly, must be influenced by, and influence, risk.

Come the 1990s, leading test professionals were striving mightily to put these insights into action. How do we integrate testing into the entire lifecycle to reduce cost and schedule impacts of bugs? How do we use risk to determine the extent and sequence of testing? How can we report test results in terms of risks mitigated and not mitigated? How can we help project teams make rational decisions regarding the optimal amount of time and energy to expend on testing?

I was one of those test professionals. So were the authors of this book, Tim Koomen, Leo van der Aalst, Bart Broekman, Michiel Vroon, and Rob Baarda. In this book, as in my own books, you can see where that synthesis has taken us. And it has taken us a long way.

As you read this book, comprehensive and comprehensible, pause to remember that it was only ten years ago that test professionals were struggling to understand and implement a complete, consistent approach to testing that managed product quality risks and delivered demonstrable business value. Now, test professionals have a number of fully articulated strategies such as those in this book and my own books which we can use to ensure that we are solving the right testing problems.

In this book, you'll find practical ideas for business driven test management and product risk analysis. These two concepts are so central to testing, yet so often ignored and misunderstood. How does testing serve the needs of the organization and project? What are the risks associated with the system under test that can and should be mitigated through testing? These may seem like obvious questions, yet all too often test teams don't know the answers, or, worse yet, they "know" the *wrong answers*. Getting the right answers to these questions is foundational to a good test effort, and serves to set the direction.

As essential as this direction-setting, foundation-laying material is, there is a lot more to this book than just a means to get the starting point right. Indeed, the authors promise a "full description of the total test process," an explanation of the entire TMap method, from start to finish, and they deliver.

Let me mention a few high points.
The chapter summarizing test design techniques is alone a good reason to have this book close at hand if you work as a test engineer. The discussion on bug management is a fine one. There is a wide-ranging discussion of estimation techniques. There is an intelligent discussion of test metrics, one that doesn't make the all-too-frequent mistake of starting with fancy Excel graphs and tables, but rather by discussing what kind of information that management needs from us as testers.
Pick a random spot in this book and you'll find something interesting. Whether you are in the first days or the final days of a test effort, you'll find pertinent ideas in this book. Like my own books, this book was obviously written to sit on your desk, ready to serve as a helpful guide on a regular basis, not to collect dust on your bookshelf.
I have in the past complained that as testers we have not built on the foundations of our profession as well as our brothers and sisters in programming. In fact, this has been a key driver for me as an author of testing books. Similarly, Tim, Leo, Bart, Michiel, and Rob have done a fine job of summarizing in this book a number of foundational concepts in software testing. For that reason, I recommend that this book join the set of ready references available to you as together we practice and improve our profession of software testing.

Rex Black, October 2006
Author of *Managing the Testing Process, Critical Testing Processes, Foundations of Software Testing,* and *Pragmatic Software Testing*
President of Rex Black Consulting Services, Inc.
President of the International Software Testing Qualifications Board

Foreword by Luc-François Salvador

It is a great privilege for me to recommend TMap Next to you as a member of the continued growing group of people who are professionally involved in improving the quality of a wide range of business processes.

Test Management Approach (TMap) was published in 1996 as a revolutionary management approach for structured testing. Within a few years, TMap was adopted worldwide by companies searching for a structured way to improve their information systems. The objective is to preserve their business processes and market image from damage.
Today TMap has become the de facto world standard for structured testing.

In the meantime, information technology is developing with an astonishing speed. The complexity of chains of information systems supporting the business is increasing rapidly. Defect tolerance is decreasing at the same speed. Experience gained by practice in the area of software testing and the knowledge of IT issues acquired over the years in professional testing are brought together in this complete update: TMap Next.

This impressive book is an absolute **must** for the modern business manager, IT manager, and Testing expert. Not only will TMap Next inform you as a manager of how to take advantage of the latest developments in the testing profession, but also, even more importantly, assist you in improving the quality of IT incorporated in your current business processes.

Without hesitation I can say this to the reader – if you practice what you read in this book the dream of predictable software engineering to realize business value is indeed attainable by your organization.

Luc-François Salvador
CEO Sogeti SA
Paris

Foreword by the authors

Introducing a completely revised and updated TMap

Ten years after the first (Dutch) book; five years after the second version; and three years after the English translation, we started with a large-scale renewal of the method at the end of 2005. The result is this new book.

We published this completely new version for several reasons:

- Over time many people have asked us to update the method and suggested ideas.
- In 2006, most organisations recognize the importance of testing and are focusing more on the duration, quantity and objects of testing.
- In the previous version, testing was described as an autonomous process in the (waterfall) development of new information systems. Current IT trends are much more broadly based: more maintenance than new development, many package implementations and iterative and agile system development. While the method evolved along with practice, the book did not. In this new version, testing is presented as a much more integral part of the big picture.
- In many organisations testing is more part of mainstream activities instead of purely as a project-based activity. A variety of line organisation are possible, including complete test factories, each with their specific pros and cons. The literature devotes little attention to this aspect.
- And perhaps most importantly: testing should be perceived much more as an economic activity within IT. Time and costs, but the benefits as well, must be made clear to the client. With this information, he or she can manage test work on the basis of the required quantity of time and costs versus the benefits: insight into quality and risks, confidence in the product, and project (management) information. This part of TMap is BDTM, Business Driven Test Management, and represents the main motive of the method.

We faced a number of challenges while revising the method. The suggested revisions sometimes conflicted heavily with the awareness that many organisations are already using TMap to their full satisfaction and are not very happy about changing their methods. We therefore decided to leave the main components of TMap intact, but incorporate the necessary revisions to the details. We proceeded with the greatest possible care. For you, as our reader, this means that you will recognise many things. We preserved the outline of the process descriptions on the basis of phases, with various activities per phase, and the attention devoted to techniques, organisation and infrastructure. We made several supplemental changes to these elements.

The main changes are:

- BDTM has been traced through the process as a motive to offer the client as many management options for testing as possible.
- The performance of a product risk analysis is described in detail.
- We have included various estimating techniques for testing.
- The management activity has been expanded considerably.
- Setting up and managing the infrastructure has become a separate phase, and we added the new role of test infrastructure coordinator.
- The description of the test design techniques has been improved significantly and updated. The techniques are now related to various coverage types.
- Several supporting processes, such as the permanent test organisation, but also the selection and implementation of tools and management of test environments, are discussed.
- Test types for regression, usability, performance, portability and security testing have been added.
- The method is described in a much wider context than simply new development in a waterfall programme.
- The entire book has been enriched with tips, more detailed information, and practical examples (more than 400).

The new TMap can be summarised by four essential points:

- It is based on a business driven test management approach that enables the client to manage the test process on rational and economic grounds.
- It provides a full description of the total test process.
- TMap contains a complete 'tool set', i.e. technique descriptions, checklists, procedures, and so on.
- It is an adaptive method that can be applied in a versatile manner, making it suitable for all test situations in most development environments, such as new development, maintenance, waterfall / iterative / agile development, customised or package software.

TMap offers the tester and the test manager guidelines to deliver results for the client.

Since the request for renewal came from both Dutch-speaking countries and other countries, the book is published concurrently in both Dutch and English.

We created this new edition without involvement from the original authors, Martin Pol, Ruud Teunissen and Erik van Veenendaal. We are still highly appreciative of their pioneering work in creating a complete test method. TMap would not be what it is without them.

Clearly a large group of people assisted in the creation of this book. Their contributions range from suggesting ideas, sharing experiences and

experience products and reviewing chapters, to helping us establish the required preconditions. Whatever their contribution, it proved vital to the quality we achieved. We would like to take this opportunity to thank them. While aware of the risk of forgetting one or more of our valued contributors, we would still like to list their names here.

We first thank the external reviewers who invested their great knowledge, time and attention to reviewing this work. These are:
Jan Blaas (KPN)
Jan Boerman (ICTRO, Ministry of Justice)
Jarmila Bökkerink-Scheerova (Philips)
Heidi Driessen (Rabobank)
Bart Dooms (Acerta, Belgium)
Ed van der Geest (SNS Bank)
Erwin Kleinveld (Rabobank)
Sander Koopman (Fortis Bank)
Ine Lutterman (Interpay)
Marco Maggi (Ministry of Agriculture, Nature and Food quality)
Benjamin Makkes van der Deijl (PGGM)
Jan Mellema (CIP Police)
Remco Möhringer (Reaal Insurance)
Paul van der Molen (Cordares)
Brian Taylor (FortisBank)
Ubel van Tergouw (UWV)
Sylvia Verschueren (ABN AMRO)
Hans Vedder (Friesland Bank)
Dennis van Velzen (AFAS ERP Software BV)

Naturally our big group of Sogeti colleagues proved immensely helpful. Their contributions vary from membership of the sounding board group to reviewing, contributing ideas and even complete texts:
Richard Ammerlaan, Eric Begeer, Paul Bentvelzen, John Bloedjes, Martin Blokpoel, Hester Blom, Martin Boomgaardt, Raoul Gisbers, Frank Goorhuis, Guy Holtus, Bart Hooft van Iddekinge, André Huikeshoven, Marco Jansen van Doorn, Ralph Klomp, Rob Kuijt as guest author of chapter 7, Peter van Lint, Dominic Maes (Sogeti Belgium), Willem-Jan van der Meer, Henk Meeuwisse, Gerrit Mudde, Guido Nelissen, Bert Noorman, André van Pelt, Elisabeth Reitsma, Ewald Rooderijs, Rob Smit, Gert Stad, Marc Valkier, Thomas Veltman, Johan Vink, Ben Visser, Harm de Vries, expertise group for usability (Kinga Visser, Gina Utama, Ronald Oomen, Wolter van Popta, Robin Klein, Thomas Veltman, Mans Scholten, Mark Schut, Martien Adema, Jeroen Bultje)

Our Sogeti USA colleagues Craig Mayer, Camille Tetta Costanzo and Dan Hannigan reviewed the English language chapters in little time and, where necessary, suggested improvements.

This work would not have been impossible without the support of the Software Control management team. We would like to express our thanks to Wim van Uden and Mark Paap in particular. Their appreciation and interest proved a huge stimulus.

We worked on the book with great enthusiasm. Whether you are an experienced TMap user or an inexperienced tester, we hope this work offers you a clear story on how to best set up the test process in all its aspects and inspire you with many ideas, detailed information and tips. We are proud of the result and hope you, our reader, will agree.

Tim Koomen
Leo van der Aalst
Bart Broekman
Michiel Vroon
Rob Baarda (project leader and editor)

Rotterdam, October 2006

Recommendations

"The quality of our IT is essential for our business operations. New releases may not disrupt the business processes, they must be faultless. To demonstrate this quality of IT and to give assurance to the business good and structured test processes are very important. This requires a fruitful cooperation between business and IT-department. I would like to stress the importance of Business Driven Test Management as explained in this book. We are implementing BDTM now step-by-step. The TMap method is an important tool for our test processes and we are glad that we as Fortis have contributed to this new version of the book."
Gert Heslenfeld,
Sectormanager IS / Accountmanager Information Services,
Fortis Insurance, The Netherlands

"TMap Next, is definitely a must read for test organizations using TPI Model for Test Process Improvement and/or TMap approach to testing. At the same time, the book is not dependent on understanding of these models, and is a useful resource for organizations implementing risk based test strategy in their organizations.. The process, techniques and tools described in this book will enable a test professional to better balance the test cost with the benefits provided by testing, thus making it easier to get management buy in for the test project."
Ruku Tekchandani
SW Validation CoP Program Manager
Intel Corporation, United States of America

"In many organisations all over the world TMap is the standard for testing for many years. The experiences gained using TMap over all these years and the new approaches in software development are the motivators to update TMap and to write this book. For instance, product risk analysis and business driven test management are added to TMap as well as use case testing and exploratory testing. The "new" TMap standard is very useful to solve the testing problems of today and tomorrow."
Prof. Dr. Andreas Spillner
University of Applied Sciences Bremen
Member of the German Testing Board (SIGIST)
Germany

"This is testing in a book. If you need to implement the practice of testing, this is a tool that is essential. Tim Koomen and colleagues have taken a great and popular book, TMap®, and have improved it in its updating – TMap Next highlights testing as an economic activity within IT and, the new emphasis on Business Driven Test Management empowers the business to get the economic and other benefits from IT that they have only dreamed of until now. This book will become a classic on software testing."
Wayne Mallinson and Peter Sage
Communications and Technical Directors respectively – Test and Data Services (Pty) Ltd
Founder and current Chairman respectively – CSSA Special Interest Group in Software Testing (SIGiST)
Gauteng South Africa

"TMap contains practical, proven ideas and methods for a risk based testing approach. The updated TMap is an important read for anyone endeavoring to understand how business driven test management fits within the context of the end-to-end process of business driven development."
Dr. Daniel Sabbah
General Manager, IBM Rational Software
United States of America

"The Ministry of Agriculture, Nature and Food Quality spends a lot of money each year to maintenance, package implementations and development of software. Standardisation at all disciplines in the software lifecycle is necessary to manage these processes. In the lifecycle, testing is becoming more important from a quality and cost perspective. That is why the Ministry has chosen for a standard method and a permanent test organisation. It is good to see this standard TMap evolving, not only adapting to the fast technological developments, but, with Business Driven Test Management, also answering to the increasing demands of clients. "
Pieter C.A. Arends
Manager Project management, Consultancy & Test services
Member of the board of IT-operations
Ministry of Agriculture, Nature and Food Quality The Netherlands

"I like this book because of it being a detailed recipe for how to organize, plan, execute and control testing. Most techniques are explained in detail, with lots of examples. Especially useful are the chapters about handling risk, and the one about test case design. The latter one is great for system and acceptance testing, while most other literature concentrates on lower level testing."
Hans Schaefer
Independent test consultant
Leader of the Norwegian Testing Board (SIGIST)

"TMap is one of the important instruments to assure the quality of our IT products. That is why REAAL Insurances has gladly contributed to the part on development testing in TMap Next."
Nico Jongerius
Member of the board/CIO of REAAL Insurances
The Netherlands

"TMap is for testing what ITIL is for operational processes. This book is a comprehensive yet highly accessible guide for both test managers and test professionals. It is truly a meaningful help to anyone who's involved in the testing business; whether you have one or more specific questions, or if you are in search of a whole testing framework or well targeted process improvements."
Wim Waterinckx
process manager testing
KBC Group
Belgium

"TMap is the most thorough to help organise testing in large IT companies".
Dorothy Graham, Grove Consultants
Great Britain

"I've been a heavy user of TMap since the availability of the book in English. I have utilized TMap for testing process communication, competence development and expanding the "skill-toolbox" of my testers in Nokia. The basic framework is very similar, but it is now much more useful because of the explicit support for wider selection of contexts.
I like the clarity of the big picture of test design: how test planning includes risk-based test strategy that links to test types, and how eventually various test design techniques provide the right depth for testing at expected coverage. As a bonus the framework suits well to expanding with your own techniques and test types."
Erkki Pöyhönen
TietoEnator Quality Assurance Competence Center, previously with Nokia
Finland

"As the quality assurance was moving from a project issue to an issue for the company board our test organisation needed to unify and structure the test process. TMap Next follows the projects phases and has as a logical toolbox. This helped us to complement our existing test process to meet the company board's timescale. TMap Next also gave us useful information how to handle people, tools, infrastructure and the organisation that in many ways has influenced the organisation of the test centre at the Swedish Tax Agency."
Henrik Rylander
Head of Test Centre
Test Unit
The Swedish Tax Agency
Sweden

Contents

PROCESSES

1 Introduction

TMap is Sogeti's prominent Test Management approach to the structured testing of information systems. The approach is described generically, since the specific makeup of this, the best fitting test approach, depends on the situation in which it is applied.
TMap can be summarised in four essentials:
1. TMap is based on a business driven test management approach.
2. TMap describes a structured test approach.
3. TMap contains a complete tool box.
4. TMap is an adaptive test method.

The first mentioned essential is directly related to the fact that the importance of the IT-business case (the justification of a project) for organisations is continuously growing. For testing this means the choices on what risks to cover with testing, what results are to be delivered and how much time and money to spend need to be made based on rational and economic grounds. This is why TMap has developed the business driven test management approach, which can be seen as the 'leading thread' of the structured TMap test process.

By describing a structured test process (essential 2) and giving a complete tool box, TMap answers the classic questions what/when, how, what with and who. With the description of test process use has been made of the TMap life cycle model: a development cycle related description of the test cycle. The life cycle model describes *what/when* should be carried out.

Besides this, to be able to execute the test process properly, several issues in the field of infrastructure (*what with*), techniques (*how*) and organisation (*who*) should be implemented. TMap provides a lot of applicable information in the shape of examples, checklists, technique descriptions, procedures, test organisation structures and test environment/tools (essential 3).

Furthermore, TMap has a flexible design so that it can be applied to several system development situations: i.e. new development as well as maintenance of information systems, development in-house or a purchased package and with outsourcing of (parts of) the testing (essential 4).

In this chapter, a sequential overview is given of how TMap became a standard approach to structured testing, the reasons for a new version of TMap, the key points of TMap and a number of suggestions concerning which chapters are of interest to which target groups.

1.1 The history of TMap

In the international testing world, TMap is a familiar concept, as this approach to testing has been in existence for some time. While it is not necessary to know the history of TMap in order to understand or apply the approach, this section invites you to take a glimpse behind the scenes.

Standard approach to structured testing

This book was preceded by the Dutch book "*Testen volgens TMap*" (= Testing according to TMap) in 1995 and "*Software Testing*" [a guide to the TMap approach] in 2002 (both books by Pol, Teunissen and Van Veenendaal)[1]. The books turned out to be, and still are, bestsellers. TMap has evolved over recent years to become a standard for testing information systems. It is currently applied in hundreds of companies and organisations, including many banks, insurance companies, pension funds and government organisations. The fact that TMap is seen to be a prominent standard approach to structured testing is demonstrated by facts as, for example, suppliers of test tools advertising with the words "applicable in combination with the TMap techniques"; test professionals who ensure that TMap experience is prominent on their CV or, increasingly, that they are TMap-certificated; recruitment advertisements in which test professionals with knowledge of TMap are sought, and independent training institutions that provide TMap courses. It is also worth mentioning the publication in a leading Dutch IT-magazine "Computable" of 30 September 2005, showing TMap to be the most-requested competence! Ahead of Java, ITIL and Unix, for example.

Moreover, its use is growing fast in other market segments, such as in the embedded software industry. Since testing in this industry differs from testing in the administrative world, another TMap concept-based book has been written especially for this [Broekman, 2003].

The strength of TMap

The strength of TMap can largely be attributed to the considerable practical experience that is the basis for the method. This experience comes from hundreds of professional testers in as many projects over the last twenty years! The aim of this book is to be a valuable aid in coping with most, if not all, challenges in the area of testing now and in the near future. The disadvantage of a book is that the content, by definition, is static, while in the field of IT new insights, system development methods, etc., are created with great regularity.

It would be commercially irresponsible to compile and publish a new version the TMap book with every new development in IT. To enable TMap to keep

[1] Up-to-date information about the translations in other languages can be found at www.tmap.net

up with current developments, an expansion mechanism is created. An expansion describes the way TMap has to be applied in a new development. Recent expansions, in the form of white papers, are included in the book *TMap Test Topics* [Koomen, 2005]. To keep the TMap users updated about the new expansions, different means of communication are used. For example, the large number of presentations and workshops that are given at test conferences, the popular TMap Test Topics sessions (in which current test themes are presented) and the many articles in the various specialist publications. All of this makes TMap what it is now: "A complete test approach, with which any organisation can successfully take on any test challenge, now and in the future!"

Tip

Take a look at www.tmap.net. You will find there, among other things:
- downloads (including white papers for expansions, checklists, test-design techniques and a glossary)
- published TMap articles
- TMap newsletters (if you wish to receive these automatically, you can register for this on the site).

1.2 TMap evolves in step

"Why then a new version of TMap," you may ask, "Was there something wrong with the previous version?" No, however, since it appeared on the market, various new developments have taken place both in IT and testing. And in order to keep TMap as complete and up to date as possible, these have been incorporated in the new version. This concerns developments such as the increasing importance of IT to organisations and a number of innovations in the area of system development. Besides this, the tester appears to be better served by a test approach written as a guide, rather than as a testing manual. An explanation of this is given in the following sections.

Increasing importance of IT to organisations

Since the end of the nineties, the use of software or information technology in general, has become increasingly important to organisations, so that IT projects are more often initiated and managed from within a user organisation. This is prompted by the following developments:

- Cost reductions of IT development and management
  IT is required to be cheaper and the business case (the 'why' in combination with costs and profits) formulated more clearly.
- Growing automation of business processes

More and more business processes within organisations are either automated or strongly dependent on other automated processes.

- Quicker deployment of automation
  With the growth of automation, IT has grown from being a company support resource to one that differentiates the company from the competition. This means that the flexibility and speed with which this resource can be deployed is of crucial importance in beating the competition.
- Quality of automation is becoming more important (see the practical example "Consequences of software failures"). The fact that IT end users currently make sure they have their say, combined with the fact that CEO's, CFO's and CIO's are made personally responsible for the accuracy of the company's financial information leads to (renewed) interest in quality of IT.

These four aspects are summed up as "more for less, faster and better". A consequence of this is that IT projects are becoming increasingly dynamic and chaotic in nature. This can put great pressure on the testers and increase the relative share of testing within IT. In order to make the test process manageable and to keep it that way, the "business driven test management" (BDTM) approach was developed for TMap, and has been incorporated in this book. With this, the creation of a test strategy is directly related to the risks. It enables the client to make responsible, risk-based choices in the test process. By making these choices, the client has significant influence on the timeline and the costs of the test process.

Real-world examples

Consequences of software failures
Because IT has become more important within organisations, the impact of any software problems is increased.
Some examples are:
- Revenue loss
- Brand/reputation loss
- Compensation claims
- Productivity loss.

The fact that this can involve considerable losses is demonstrated by the following real-world examples:
- A sporting goods manufacturer suffered a 24% drop in turnover (€100 million) in a single quarter because of software faults in the stock administration. In the days that followed, after the company had announced that they had problems with the software, the share value decreased by more than 20%.
- Through a fault in their encryption software, a financial advisory organisation displayed their customers' social security numbers and passwords in legible text on

their website. This caused distress among the customers and led to a sizeable loss of business.
- After a pharmaceutical wholesale company had gone bankrupt, the parent company submitted a claim for damages of €500 million to the software supplier, claiming that the 'enterprise software' had been faulty.

Innovations in the system development area

The test principles on which TMap is based, came into being in the eighties, and, of course, they relate to the system development methods of that time. Nowadays, these methods are often referred to as 'waterfall' methods. Their most important characteristic is the purely sequential execution of the various development activities. Besides that there is a growing interest in other methods. The most important characteristic of the new generation of system development methods is incremental or iterative development, with the test process increasingly being integrated in the development process. Dynamic Systems Development Methodology® (DSDM), Rational Unified Process® (RUP), Rapid Application Development (RAD) and the Agile approach are examples of this.

Since, in the course of various activities, testing touches on the chosen system development method, it is inevitable that a test approach like TMap should evolve in step with changes in the area of system development. How to apply TMap in certain situations is often already laid down in the previously mentioned expansions. The key points of the expansions have now been integrated in this new version of TMap, so that the book again provides a current and as complete as possible overview of how TMap can be applied in a variety of situations.

Test approach guide

The previously mentioned two developments, i.e. the increasing importance of IT within organisations and the innovations in system development, indicate the increased dynamic of the various development and test environments. In a situation like this, a manual is often seen as being too rigid. Furthermore, it appears that testing is increasingly being carried out by a broader public with a wide range of competencies, so that there is now a greater need for more comprehensive descriptions of how specific TMap activities can be carried out. A consequence of this for the description of the test approach is that it needs to be written more as a guide than a manual, taking the reader 'by the hand'.

The version of TMap now before you has been entirely revised in response to the developments mentioned and to the many requests for amendments and amplifications to the book.

In short, this version of TMap *offers the tester a guide to delivering results to the client.*

1.3 What TMap offers

But what exactly does TMap now offer – what can you do with it? You will find the answer to that question in this section, in which a general overview is given of the assistance that TMap can supply and where TMap can be applied. In the rest of the book, this is gone into in more detail and much attention is paid to the way in which TMap can be applied in various situations.

1.3.1 Where TMap helps

In order to assist the tester in his work, TMap explains how to carry out certain activities, or how these are supported by TMap. This concerns help with:

- the translation of the client's requirements into a concrete test approach and management of the execution
- assisting the test manager, test coordinator and/or tester to deal with the various IT development approaches, each from within their own areas of responsibility
- the execution of, among other things:
  - a product risk analysis
  - a test strategy
  - a (non-)functional test
- the organisation of testers in e.g. an existing test organisation or in flexible project teams
- the setting up and the management of test infrastructure for the current and other projects
- the creation of test designs and the use of various test design techniques
- the preparation, specification and execution of the tests, described as processes within the TMap processes
- the execution of the test activities with real-world examples, tips and also detailed explanations of certain aspects
- the reporting of the test results from the perspective of the client
- supplying information in connection with the project result
- considering the test process as much as possible from the exterior vantage point, by answering, for example, practical questions (what does testing actually deliver?) and making use of general project information
- selecting the right test dialogue partner for specific clients:
  - a test manager views the testing from a wider perspective (focusing on the environment) and is therefore a suitable contact for clients at management level

- a test coordinator focuses mainly on the internal test process and will therefore often be the contact for e.g. project leaders
- a tester is task-focused and will concern himself mainly with the design and execution of the test cases.

1.3.2 Where TMap can be applied

These days, the IT development approach is extremely variable. TMap addresses this directly and first defines the following possibilities of applying TMap:

- where there is either a demand-supplier relationship (e.g. outsourcing) between client, developer and tester (each with their own responsibilities), or a collective interactive approach
- with iterative, incremental, waterfall and agile approaches
- with new development, maintenance and migration of information systems
- in situations with combinations of development approaches, such as in-house, reuse-based, use of standard packages and assembling of purchased modules, all within a single IT architecture
- with coverage of non-functional requirements of the information system in the test approach
- in situations where much attention has to be paid to the communication process and associated skills.

Tip

The implementation or improvement of a test approach is not something that can be done casually. Among other things, it requires knowledge of the current test maturity of an organisation and of the environment in which the testing is to be implemented or improved. It should also be clear why there is a need for certain test aspects to be improved. It often seems difficult in practice to determine what steps should be taken in which sequence in order to implement or improve the testing. The “Test Process Improvement” model [Koomen, 1999] is a popular model of step-by-step implementation and improvement of a test approach.

1.4 Reading guide and the most important changes

If you are reading this book,[2] the chances are that you want an answer to a specific (test) question. If you don't wish to read the whole book, the following description of its structure and the reading guide should lead you to the answer quickly.
To notice all the differences compared with the last version, the whole book should be read. Most of the text has been re-written and the book contains more than 400 new ideas, tips and examples. We can however, imagine that an experienced TMap user would quickly like to find the changes made. Therefore we have added a section that refers to the chapters, sections and subjects which have undergone major, important changes (section 1.4.3).

1.4.1 Structure of the book

The book has three parts: general, processes and components.

The general part describes the framework and importance of testing. Besides giving the history and evolution of TMap, it also describes what TMap has to offer, providing a summary overview.

The processes part describes the test processes, including the master test plan (managing the total test process), acceptance and system testing and development testing. Also described in this part are the associated supporting processes (including the permanent test organisation and the organisation and management of test environments).

The components part describes a number of components that can be used independently (including quality characteristics and test roles) or as an aid in the processes (including product risk analysis and test design techniques). For a complete overview of these components, refer to table 1.1 "Reading suggestions" in the next section.

At various places in the book, definitions, practical examples, tips and more detailed explanations are provided. These can be recognised by title and a box or a grey background.

[2] For the sake of readability of the book, the use of he/she has not been included. However, where reference is made to 'he', it should be understood that 'she' is emphatically also intended.

1.4.2 Reading guide

The main target group of this book is those who are directly involved in the test process. A test team can use the book as a guide for carrying out the test activities. For those who operate further on from the primary test process, such as clients, end users and EDP auditors, the book offers good insight into professional testing. To this end, some chapters have been included on the background and organisation of testing.

"TMap Next" is not a book that asks to be read from cover to cover. Depending on your involvement in testing, chapters can be read carefully, scanned quickly, or even skipped altogether. The requirement for, or interest in, the various chapters will vary per target group. As an aid to choosing the chapters that are of interest in connection with certain questions or target groups, reading suggestions have been provided (table 1.1). On the next page we distinguish some possible questions and target groups:

A. You are requested to set up the testing of an application and to manage its execution (test managers, test coordinators, etc.)
B. You are requested to test an application (testers, developers, users and system administrators)
C. You are requested to assess the quality of a test with associated processes and products (EDP auditors and quality assurance employees)
D. You are responsible for the IT department or are the owner of an application and wish to have professional testing carried out (development and test process clients and relevant line management)
E. With your training background, you are interested in the testing profession (students and tutors of information technology/business economics)
F. You are responsible for HRM within the organisation and you are recruiting testers (personnel & organisation employees).

Chapter		A	B	C	D	E	F
General							
Ch01	Introduction	√	√	√	√	√	√
Ch02	Framework and importance of testing	√	√	√	√	√	√
Ch03	The essentials of TMap	√	√	√	√	√	√
Processes							
Ch04	Introduction to the processes	√	√	√	√	√	
Ch05	Master test plan, managing the total test process	√		√	√	√	
Ch06	Acceptance and system tests	√	√	√	√	√	
Ch07	Development tests	√	√	√			
Ch08	Supporting processes	√			√		√
Components							
Ch09	Product risk analysis	√					
Ch10	Quality characteristics and test types	√	√				
Ch11	Estimation techniques	√					
Ch12	Defects management	√	√				
Ch13	Metrics	√					
Ch14	Test design techniques		√			√	
Ch15	Evaluation techniques	√	√				
Ch16	Test roles	√					√

Table 1.1: Reading suggestions

Table key: √, the relevant target group is advised to study the chapter well. The other chapters are optional reading (this is certainly recommended to group E).

1.4.3 The most important changes

The changes within TMap have been integrated with all the chapters of the book. It is therefore almost impossible to pinpoint those exact places in the book that have been changed. We are convinced that an experienced TMap user can fathom the most important changes without reading the entire book. For them we have drawn up the list below referring to all chapters, sections and subjects that have undergone important changes:

- TMap is divided in four essentials. Chapter 3 gives a full description of this subject. It also contains an overview of business driven test management.
- A full description of implementing a product risk analysis is given in chapter 9.
- Business driven test management and the product risk analysis have a great influence on, amongst others, understanding the testing assignment, strategy, estimation and planning. See chapter 5 and 6.
- Control phase of the total test process and the Control phase for acceptance and system testing have become separate phases (separated from the Planning phase). See sections 5.3 and 6.3.
- The Setting up and maintaining infrastructure phase is new. See section 6.4.
- The Planning phase and the Control phase (of the total test process) mainly describe the test management activities. To be able to see all the changes (shifts) the activity diagrams in chapters 5 and 6 should be looked at.
- Chapter 11 contains a great number of new estimation techniques.
- A new 'look' on development testing is described in chapter 7.
- Besides the test process chapters we now have a chapter that contains supporting processes. This chapter 8 goes further into subjects such as: test policy, permanent test organisation, test environments, test tools and the test professional.
- In chapter 14, mainly in the first three sections, a new angle can be found on coverage, basic techniques and test design techniques.
- The test design techniques decision table test and data combination test have been altered. New techniques are use case test and exploratory testing. Besides these four techniques section 14.4 describes a further seven techniques.
- The set of test types have been extended with testing of regression, usability, performance, portability and security in section 10.3.

2 Framework and importance of testing

This chapter provides an introduction to testing in general and focuses on structured testing. No specific (prior) knowledge of TMap is required in order to understand this. In sequence, an explanation is given of: what is understood by testing, why testing is necessary (and what it delivers), what the role of testing is and what structured testing involves.

In chapter 3 “The essentials of TMap”, a description is given of a structured testing approach using TMap.

2.1 What is testing?

While many definitions of the concept of testing exist, one way or another they all contain comparable aspects. Each of the definitions centres on the comparison of the test object against a standard (e.g. expectation, correct operation, requirement). With this, it is important to know exactly what you are going to test (the test object), against what you are going to compare it to (the test basis) and how you are going to test it (the test methods and techniques).

The International Standardisation Organisation (ISO) and the International Electrotechnical Commission (IEC) apply the following definition [ISO/IEC, 1991]:

Definition

Testing is a technical operation that consists of the determination of one or more characteristics of a given product, process or service according to a specified procedure.

Testing supplies insight in the difference between the actual and the required status of an object. Where quality is roughly to be described as ‘meeting the requirements and expectations’, testing delivers information on the quality. It provides insight into, for example, the risks that are involved in accepting lesser quality. For that is the main aim of testing. Testing is one of the means of detection used within a quality control system. It is related to reviewing, simulation, inspection, auditing, desk-checking, walkthrough, etc. The various instruments of detection are spread across the groups of evaluation and testing[1]:

[1] The theory also refers to verification and validation. Verification involves the evaluation of (part of) a system to determine whether the products of a development phase meet the conditions that were set at the beginning of that phase. Validation is understood to mean determining whether the products of the system development meet the user needs and requirements. [IEEE, 1998]

- Evaluation: assessment of interim products.
- Testing: assessment of the end products.

Put bluntly, the main aim of testing is to find defects: testing aims to bring to light the lack in quality, which reveals itself in defects. Put formally: it aims to establish the difference between the product and the previously set requirements. Put positively: it aims to create faith in the product.

The level of product quality bears a relationship to the risks that an organisation takes when these products are put into operation. Therefore, in this book we define testing, according to TMap, as follows:

Definition

Testing is a process that provides insight into, and advice on, quality and the related risks.

Advice on the quality of what? Before an answer to this can be given, the concept of quality requires further explanation. What, in fact, is quality?

Definition

The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs [ISO, 1994].

In aiming to convert 'implied needs' into 'stated needs' we soon discover the difficulty of subjecting the quality of an information system to discussion. The language for discussing quality is lacking. However, since 1977, when McCall [McCall, 1977] came up with the proposal to divide the concept of quality into a number of different properties, the so-called quality characteristics, much progress has been made in this area.

Definition

A quality characteristic describes a property of an information system.

A well-known set of quality characteristics was issued by the ISO and IEC [ISO 9126-1, 1999]. In addition, organisations often create their own variation of the above set. For TMap, a set of quality characteristics specifically suited to testing has been compiled, and these are listed and explained in chapter 10, "Quality characteristics and test types". This set is the one that is used within the framework of this book.

What, then, is the answer to the question: "Advice on the quality of what?" Since, where quality is concerned, the issue is usually the correct operation of the software, testing can be summed up as being seen by many to mean:

establishing that the software functions correctly. While this may be a good answer in certain cases, it should be realised that testing is more than that. Apart from the software, other test objects exist, the quality of which can be established. That which is tested, and upon which quality recommendations are subsequently given, is referred to as a test object.

Definition

The test object is the information system (or part thereof) to be tested.

A test object consists of hardware, system software, application software, organisation, procedures, documentation or implementation. Advising on the quality of these can involve – apart from functionality – quality characteristics such as security, user friendliness, performance, maintainability, portability and testability.

Pitfalls

In practice, it is by no means clear to everyone what testing is and what could or should be tested. Here are a few examples of what testing is *not*:

- Testing is not a matter of releasing or accepting something. Testing supplies advice on the quality. The decision as regards release is up to others (stakeholders), usually the commissioner of the test.
- Testing is not a post-development phase. It covers a series of activities that should be carried out in parallel to development.
- Testing is something other than the implementation of an information system. Test results are rather more inclined to hinder the implementation plans. And it is important to have these – often closely related - activities well accommodated organisationally.
- Testing is not intended initially to establish whether the correct functionality has been implemented, but to play an important part in establishing whether the required functionality has been implemented. While the test should of course not be discounted, the judgement of whether the right solution has been specified is another issue.
- Testing is not cheap. However, a good, timely executed test will have a positive influence on the development process and a qualitatively better system can be delivered, so that fewer disruptions will occur during production. Boehm demonstrated long ago that the reworking of defects costs increasing effort, time and money in proportion to the length of time between the first moment of their existence and the moment of their detection [Boehm, 1981]. See also “What does testing deliver?” in the next section.
- Testing is not training for operation and management. Because a test process generally lends itself very well to this purpose, this aspect is often too easily included as a secondary request. Solid agreements should see

to it that both the test and the training will be qualitatively adequate. A budget and time should be made exclusively available for the training, and agreements made as regards priorities, since at certain times choices will have to be made.

It is the task of the test manager, among others, to see that these pitfalls are avoided and to make it clear to the client exactly what testing involves.

2.2 Why test?

In chapter 1 “Introduction”, it is explained that IT has been increasing in importance to organisations since the end of the nineties. But with this, many organisations are plagued by projects getting out of hand in terms of both budget and time, owing to software defects during the operation of the developed information systems. This shows that organisations are accepting, or having to accept, systems without having real insight into their quality. In many cases, too, there is a lack of good management information upon release. This often results in big risks to the company operations: high reworking costs, productivity loss, brand/reputation loss, compensation claims and loss of competitiveness through late availability of the new product (revenue loss) may be the consequences.

Before an information system goes into production, the organisation will have to ask itself explicitly whether all requirements are met. Have all the parts and aspects of the system been explored in sufficient depth? Besides the functionality, have checks been carried out on, for example, the effectivity, performance and security aspects? Or, as ISO puts it: has it been established whether the product possesses the characteristics and features necessary to meet the stated or (even more difficult) implied needs? What is self-evidently implied to one may be a revelation to another.

Have all the errors been reworked, and have any new errors been introduced in the course of reworking them? Can the company operations depend on this system? Does the system really provide the reliable solution to the information issue for which it was designed?

The real question is: what risks are we taking and what measures have been taken to reduce those risks. In order to avoid obtaining answers to these crucial questions only at the operational phase, a good, reliable testing process is required. That demands a structured test approach, organisation and infrastructure, with which continuous insight may be obtained in a controlled manner into the status of the system and the risks involved.

What does testing deliver?

While a structured test approach is considered to be of great importance, the question "What do you think of the test process?" is generally answered with "Expensive!" This response can seldom be quantified, since it is often a gut reaction, unsupported by figures. Testing is expensive. Yes, that is true if you only look at the test costs and disregard the test benefits. Test costs are based on, among other things:

- the costs of the test infrastructure
- the hours worked by the testers and their fees.

Test benefits are [Black, 2002]:

- The prevention of (high) reworking costs and consequential damage to the production situation, thanks to defects being found during testing and rectified within the system development process. Examples of consequential damage are: revenue loss, brand/reputation loss, compensation claims and productivity loss.
- The prevention of damage in production, thanks to errors being found during testing, and, while not being solved, being flagged as 'known errors'.
- Having/gaining faith in the product.
- Facilitating good project management through the supply of (progress and quality) information.

If there is a way of expressing test benefits in money, the answer to the question "What does testing deliver" may be, from a test-economic perspective:

Test Yield = Test Benefits – Test Costs

Although this appears to be a simple calculation, in practice it is very difficult to estimate how much damage would have been caused by failures that were found during testing, had they occurred at the production stage. And anyway, how do you translate, for example, potential loss of image into money? In the literature, some attempts have nevertheless been made at making this calculation (e.g.: [Aalst, 1999]).

However, it remains difficult to establish exactly what having faith in the quality of a product, or gaining (progressive) information really delivers. Despite that, within the world of testing there are more and more tips and tricks to be found that make it possible to observe one or more of the following defects and to operate accordingly:

- too much testing is carried out, so that the costs of finding a defect during testing no longer offset the damage that this defect would potentially cause if it occurred in production
- too little testing is done, so that more issues occur in production and the

reworking costs of these are proportionately higher than the test costs would have been to find the defects during testing
- testing is carried out ineffectively, so that the test investment is not recouped.

2.3 The role of testing

This section explains both the significance and role of certain test concepts in their environment. Spread across the following subjects, the associated concepts are explained:
- Testing and quality management
- Testing: how and by whom
- Test and system development process
- Test levels and responsibilities
- Test types

2.3.1 Testing and quality management

Quality was, is and remains a challenge within the IT industry (see also examples in section 1.2 "TMap evolves in step"). Testing is not the sole solution to this. After all, quality has to be built in, not tested in! Testing is *the* instrument that can provide insight into the quality of information systems, so that test results – provided that they are accurately interpreted – deliver a contribution to the improvement of the quality of information systems. Testing should be embedded in a system of measures in order to arrive at quality. In other words, testing should be embedded in the quality management of the organisation.

The definition of quality as expressed by the ISO strongly hints at its elusiveness. What is clearly implied to one is anything but to another. Implicitness is very much subjective. An important aspect of quality management is therefore the minimisation of implied requirements, by converting them into specified requirements and making visible to what degree the specified requirements are met. The structural improvement of quality should take place top-down. To this end, measures should be taken to establish those requirements and to render the development process manageable.

Definition

Quality assurance covers all the planned and systematic activities necessary to provide adequate confidence that a product or service meets the requirements for quality [ISO, 1994].

These measures should lead to a situation whereby:
- there are measurement points and units that provide an indication of the quality of the processes (standardisation)
- it is clear to the individual employee which requirements his work must meet and also that he can evaluate them on the basis of the above-mentioned standards
- it is possible for an independent party to evaluate the products/services on the basis of the above-mentioned standards
- the management can trace the causes of weaknesses in products or services, and consider how they can be prevented in future.

These measures may be divided into preventive, detective and corrective measures:
- Preventive measures are aimed at preventing a lack in quality. They can be, for example, documentation standards, methods, techniques, training, etc.
- Detective measures are aimed at discovering a lack of quality, for example by evaluation (including inspections, reviews, walkthroughs) and testing.
- Corrective measures are aimed at rectifying the lack of quality, such as the reworking of defects that have been exposed by means of testing.

It is of essential importance that the various measures are cohesive. Testing is not an independent activity; it is only a small cog in the quality management wheel. It is only one of the forms of quality control that can be employed. Quality control is in turn only one of the activities aimed at guaranteeing quality. And quality assurance is, in the end, only one dimension of quality management.

2.3.2 Testing: how and by whom

Testing often attracts little attention until the moment the test is about to begin. Then suddenly a large number of interested parties ask the test manager about the status. This section demonstrates, however, that testing is more than just the execution of tests. We then explain the ways of testing and by whom the testing can be carried out.

There is more to testing

Testing is more than a matter of taking measurements – crucially, it involves the right planning and preparation. Testing is the tip of the iceberg, the bigger part of which is hidden from view (see figure 2.1 “The iceberg”).

Figure 2.1: The iceberg

In this analogy, the actual execution of the tests is the visible part, but on average, it only covers 40% of the test activities. The other activities – planning and preparation – take up on average 20% and 40% of the testing effort respectively. This part is not usually recognised as such by the organisation, while in fact it is where the biggest benefit, not least regarding time, is to be gained. And, significantly, by carrying out these activities as much as possible in advance of the actual test execution, the testing is on the critical path of the system development programme as briefly as possible. It is even possible, because of technical developments (test automation), to see a decreasing line in the percentage of test execution compared with preparation and planning.

Ways of testing

There are various ways of testing (in this case, executing tests). For example, is the testing being done by running the software or by static analysis? And is a characteristic of the system being tested using test cases specially designed for it, or precisely not? A number of ways of testing are:

- Dynamic explicit testing
- Dynamic implicit testing
- Static testing

Dynamic explicit testing

With dynamic explicit testing, the test cases are explicitly designed to obtain information on the relevant quality characteristic. With the execution of the test, or the running of software, the actual result is compared against the expected result in order to determine whether the system is behaving according to requirements. This is the most usual way of testing.

Dynamic implicit testing

During dynamic testing, information can also be gleaned concerning other quality characteristics, for which no explicit test cases have been designed. This is called dynamic implicit testing. Judgements can be made, for example, on the user-friendliness or performance of a system based on experience gained without the specific test cases being present. This can be planned if there has been a prior agreement to provide findings on it, but it can also take place without being planned. For example, if crashes occur regularly during the testing. In that case, a judgement can be made concerning the reliability of the system.

Static testing

With static testing, the end products are assessed without software being run. This test usually consists of the inspection of documentation, such as security procedures, training, manuals, etc., and is often supported by checklists.

Who tests?

Anyone can do testing. Who actually does the testing is partly determined by the role or responsibility held by someone at a given time. This often concerns representatives from development, users and/or management departments. Besides these, testing is carried out by professional testers, who are trained in testing and who often bring a different perspective to testing. Where, for example, a developer wants to demonstrate that the software works well ("Surely I'm capable of programming?"), the test professional will go in search of defects in the software. Moreover, a test professional is involved full-time in testing, while the aforementioned department representatives in many cases carry out the testing as a side issue. In practice, the mix of well-trained test professionals and representatives from the various departments leads to fruitful interaction, with one being strong in testing knowledge and the other contributing much subject or system knowledge.

2.3.3 Test and system development process

The test and system development processes are closely intertwined. One delivers the products, which are evaluated and tested by the other. A common way of visualising the relationship between these processes is the so-called V model. A widely held misunderstanding is that the V model is suited only for a waterfall method. But that misrepresents the intention behind the model. It is also eminently usable with an iterative and incremental system development method. Therefore, with such a method, a V model can be drawn, for example, for each increment. Many situations are conceivable that influence the shape and the specific parts of the V model.

A few situations are shown in the box below: "Influences on the V model". With the help of the V model, the correlation between test basis, evaluation and testing (test levels) is explained in this and the following subsection.

In more detail

Influences on the V model

The form and specific parts of a V model can vary through, for example:

- The place of the testing within the system development approach.
  - Using a waterfall development method with characteristics including: construction of the system in one go, phased with clear transfer points, often a lengthy cyclical process (SDM, among others).
  - Using an incremental and iterative development method with the following possible characteristics: constructing the system in parts, phased with clear transfer points; short cyclical process (DSDM and RUP, among others).
  - Using an agile development method characterised by the four principles: individuals and interaction over processes and tools, working software over extensive system documentation, user's input over contract negotiation, reacting to changes over following a plan (extreme programming and SCRUM, among others).
- The place of testing within the life cycle of the information system.
  - Are we looking at new development or the maintenance of a system?
  - Does this involve the conversion or migration of a system?
- A self-developed system, a purchased package, purchased components, or distributed systems.
- The situation whereby (parts of) the system development and/or (parts of) the testing are outsourced (outsourcing and off-/near shoring, among other things).

Left side of the V model

In figure 2.2 "V model (the left side)" the left-hand side shows the phases in which the system is built or converted from wish, legislation, policy, opportunity and/or problem into the realised solution. In this case, the left-hand side shows the concepts of requirements, functional and technical designs and realisation. While the exact naming of these concepts is dependent on the selected development method, it is not required in order to indicate the relationship between the system development and test process at a general level.

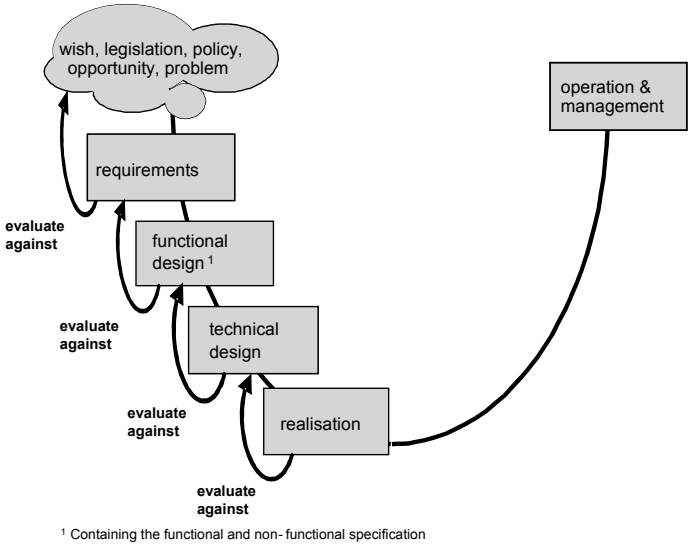

Figure 2.2: V model (the left side)

Evaluation

During the system development process, various interim products are developed. Depending on the selected method, these take a particular form, content and relationship with each other and can be tested on these.

Definition

Evaluation is assessing the interim products in the system development process.

In the V model, the left-hand side shows which interim products can be evaluated (against each other). In evaluation, the result can be compared with:

- The preceding interim product
  For example, is the functional design consistent with the technical design?
- The requirements from the succeeding phase
  For example, can the builder realise the given design unambiguously and are the specifications testable?
- Other interim products at the same level
  For example, is the functional design consistent internally and with functional designs related to it?
- The agreed product standard
  For example, are there use cases present?

- The expectations of the client (see box "Realised requirements") Is the interim product still consistent with the expectations of the acceptors?

With this, various techniques are available for the evaluation: reviews, inspections and walkthroughs (see also chapter 15 "Evaluation techniques").

In more detail

Realised requirements
What about the trajectory of wish, legislation, etc., to product? Will, for example, all the requirements be realised, or will something be lost along the way? A survey carried out by the Standish Group unfortunately shows a less than encouraging picture. The findings of the survey (figure 2.3 "Realised requirements"), in which the percentage of realised requirements was determined, shows that, of the original defined requirements, only 42% to 67% are actually realised by the project [The Standish Group, 2003].

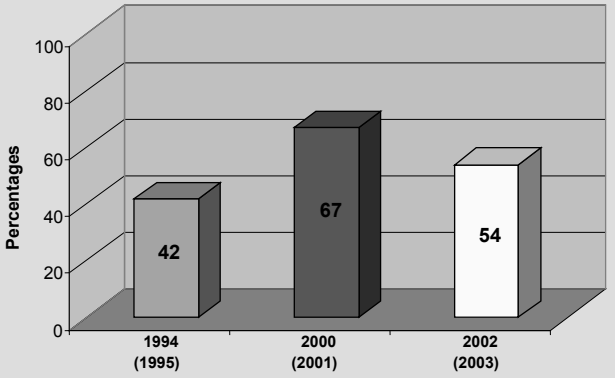

Figure 2.3: Realised requirements

Besides normal evaluation results (the finding of defects) a well-organised and executed evaluation process can deliver a contribution to a higher realisation percentage in respect of the original defined requirements.

2.3.4 Test levels and responsibilities

In a system development phase, a separation can be made between the responsibilities of client, user, manager and system administrator on the one hand and system developer and supplier on the other. In the context of testing, the first group is collectively known as the accepting (requesting) party and the second group as the supplying party. Other concepts that are

also mentioned in this connection are the demand and supply organisations. At a general level, there are two possible aims in testing:

- The supplying party demonstrates that what should be supplied actually is supplied.
- The accepting party establishes whether what has been requested has actually been received and whether they can do with the product what they want to/need to do.

Right side of the V model

In figure 2.4 "V model (the right side), a horizontal dotted line indicates this (formal) separation. In practice, the separation is less concrete and system developers will be called in for the information analysis and the setting up of the specifications, and will also supply support with the acceptance test. The expertise of users and administrators will also be employed in the building activities. It is important to define the various responsibilities clearly. This certainly applies to the testing. Who acts as the client of a test, who accepts it, who wants advice on the quality and who will test what, and when?

Testing takes place at the right-hand side of the V model. With this, a distinction is often made within the supplying party between testing by the developer and testing by the project/supplier:

- Testing by developer
  For example, by a programmer, technical designer or engineer.
- Testing by project/supplier
  For example, by a project or supplier of software or package supplier, or maintenance organisation.

In practice, this distinction in (test) responsibilities is translated into the grouping of test activities into test levels.

Definition

A test level is a group of test activities that are managed and executed collectively.

For every phase of construction, there are one or more test levels. A misconception in this is that the test level rectangles in the V model are seen as phases of the system development process. However, these represent the test execution (the measuring phase) of a test level.

Figure 2.4 "V model (the right side)" places the development and system tests under the responsibility of the supplying party and the acceptance tests under the responsibility of the accepting party:

- Development tests
  Tests in which the developing party demonstrates that the product meets the technical specifications, among other things.
- System tests
  Tests in which the supplying party demonstrates that the product meets the functional and non-functional specifications and the technical design, among other things.
- Acceptance tests
  Tests in which the accepting party establishes that the product meets expectations, among other things.

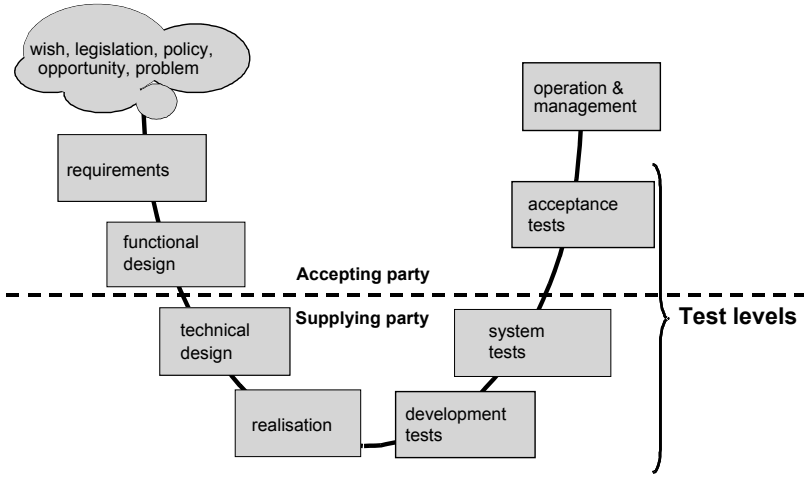

Figure 2.4: V model (the right side)

Although test level is a much-used concept, in practice people often have difficulty in substantiating it. It does not appear to be possible to designate *the* test level set. Even within one company, it is often impossible to define one set that should be used in every project.

In this book, we refer to the three test levels mentioned. In incidental cases, in order to describe a certain case appropriately, these test levels are further subdivided. In chapter 4 “Introduction to the processes”, the set of test levels used by TMap are specified more closely.

As mentioned previously, there is no such thing as one standard set of test levels. This is simply because it strongly depends on the organisation, the project and even the individual. But of course, there are some indications available for arriving at a relevant test level categorisation in a particular

situation. You can see these indications in chapter 5 “Master test plan, managing the total test process”.

Test basis and test levels

A test level is aimed at demonstrating the degree to which the product meets certain expectations, requirements, functional specifications or technical specifications. System documentation is often used here for reference. In certain situations, usually in cases of migration operations, the current production system may also serve for reference. If there is little, no, or only obsolete system documentation available, the knowledge of, for example, the end users and product owners may be used for reference. There are many sources of information that can be used for reference in testing. The general term for this is ‘test basis’.

Definition

The test basis is the information that defines the required system behaviour.

This is used as a basis for creating test cases. In the event that a test basis can only be changed via the formal change procedure, this is referred to as a ‘fixed test basis’.

Figure 2.5 “V model (test basis)” uses arrows with the text “Input for” to indicate which information sources can be used in which test level as a basis in order to derive test cases. From the model it also appears that it is possible that the same test basis is being used in two test levels. This often happens when there are two different parties carrying out a test according to their individual responsibilities. In the illustrated model, a functional design is used as test basis by the supplying party to demonstrate to the accepting party, for example, that the system complies with this design. However, the accepting party uses the same functional design as a test basis in order to check whether the system supplied actually complies with this design.

It is obvious that in such a situation, there is a chance of duplicate testing being carried out. This can be a conscious and perfectly valid decision, but it can be equally justifiable to combine certain test levels. In chapter 5 “Master test plan, managing the total test process”, you will find a number of indications for making selections in this.

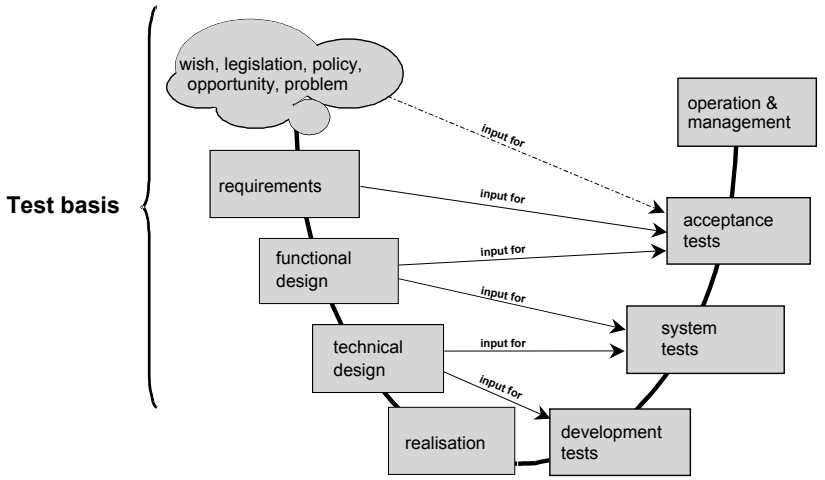

Figure 2.5: V model (test basis)

2.3.5 Test types

During the testing of a system, various types of properties can be looked at - the so-called quality characteristics. Examples of these are functionality, performance and continuity.

At detail level, it may be, however, that a certain quality characteristic is too general, making this difficult to maintain in practice. The quality characteristic should then be cast in a form that is easier for the tester to use. In the example of functionality, risks are conceivable in the area of interfaces, in- and output validations, relationship checks, or just the processing. With performance we could look to load and/or stress risks. And in the example of continuity, there is the matter of the importance of backup, restore and/or failover facilities. This commonly used form of quality characteristics is called the “test type”.

Definition

A test type is a group of test activities with the intention of checking the information system in respect of a number of correlated (part aspects of) quality characteristics.

On the website www.tmap.net, you will find a list of a number of common test types. This list is not exhaustive and will vary from test project to test project, and from organisation to organisation.

A strange fish among these is the "regression" test type: a test type was, after all, intended to provide detailed information on the specific risks with regard to the relevant quality characteristics, while regression, on the contrary, is a rather general term and in fact cites a specific risk in itself.

Definition

Regression is the phenomenon that the quality of a system deteriorates as a whole as a result of individual amendments.

Definition

A regression test is aimed at verifying that all the unchanged parts of a system still function correctly after the implementation of a change.

Often, the establishment of whether a regression has taken place is an aim in itself. It is therefore better to pay some attention to it here, where the distribution of quality characteristics across test levels is being considered in general terms.

When filling in the detail, thought could be given to what is meant by "correct functioning" in the above definition. Does this concern, for example, functionality, performance or failover facilities? In fact, all the quality characteristics and test types can be part of a regression test.

How test types are used within TMap is explained in chapters 6 "Acceptance tests and system tests" and 10 "Quality characteristics and test types".

2.4 What is structured testing?

In practice, it seems that testing is still being carried out in an unstructured manner in many projects. This section, besides citing a number of disadvantages of unstructured testing and advantages of structured testing, also cites a few characteristics of the structured approach.

Disadvantages of unstructured testing

Unstructured testing is typified by a disorderly situation, in which it is impossible to predict the test effort, to execute tests feasibly or to measure results effectively. This is often referred to as 'ad hoc testing'. Such an approach employs no quality criteria in order to, for example, determine and prioritise risks and test activities. Neither is a test-design technique employed for the creation of test cases. Some of the findings that have resulted from the various studies of structured and unstructured testing are:

- Time pressures owing to:
  - absence of a good test plan and budgeting method
  - absence of an approach in which it is stated which test activities are to be carried out in which phase, and by whom
  - absence of solid agreements on terms and procedures for delivery and reworking of the applications.
- No insight in or ability to supply advice on the quality of the system due to:
  - absence of a risk strategy
  - absence of a test strategy
  - test design techniques not being used, therefore both quality and quantity of the test cases are inadequate.
- Inefficiency and ineffectiveness owing to:
  - lack of coordination between the various test parties, so that objects are potentially tested more than once, or even worse: not tested at all
  - lack of agreements in the area of configuration and change management for both test and system development products
  - the incorrect or non-use of the – often available – testing tools
  - lack of prioritisation, so that less important parts are often tested before more risk-related parts.

Advantages of a structured testing approach

So what are the advantages, then, of structured testing? A simple, but correct, answer to that is that in a structured approach, the aforementioned disadvantages are absent. Or, put positively, a structured testing approach offers the following advantages:

- it can be used in any situation, regardless of who the client is or which system development approach is used
- it delivers insight into, and advice on, any risks in respect of the quality of the tested system
- it finds defects at an early stage
- it prevents defects
- the testing is on the critical path of the total development as briefly as possible, so that the total lead time of the development is shortened
- the test products (e.g. test cases) are reusable
- the test process is comprehensible and manageable.

Features of the structured testing approach

What does the structured testing approach look like? Many different forms are conceivable. In chapter 3 “The essentials of TMap” and the subsequent chapters, the specific TMap form of this is given.

In general, it can be said that a structured testing approach is typified by:

- Providing a structure, so that it is clear exactly *what*, by *whom*, *when* and in *what sequence* has to be done.
- Covering the full scope and describing the complete range of relevant aspects.
- Providing concrete footholds, so that the wheel needn’t be reinvented repeatedly.
- Managing test activities in the context of time, money and quality.

3 The essentials of TMap

This chapter describes the specific TMap content of a structured test approach. The content can be summarised in four essentials.

The four essentials of TMap

1. TMap is based on a business driven test management (BDTM) approach.
2. TMap describes a structured test process.
3. TMap contains a complete tool box.
4. TMap is an adaptive test method.

The first essential can be related directly to the fact that the business case of IT is becoming ever more important to organisations. The BDTM approach provides content that addresses this fact in TMap and can therefore be seen as the 'leading thread' of the structured TMap test process (essential 2). The TMap life cycle model is used in the description of the test process. Furthermore various aspects in the field of infrastructure, techniques and organisation must be set up to execute the test process correctly. TMap provides a lot of practical applicable information on this, in the form of e.g. examples, checklists, technique descriptions, procedures, test organisation structures, test environments and test tools (essential 3). TMap also has a flexible setup so that it can be implemented in different system development situations: both for new development and maintenance of a system, for a self-developed system or an acquired package, and for outsourcing (parts of) the testing process. In other words, TMap is an adaptive method (essential 4).

In figure 3.1 "TMap model of essentials", the left triangle symbolises BDTM, the triangle at the bottom the tool box, the parallelogram the structured test process, and the 'circle' TMap's adaptiveness.

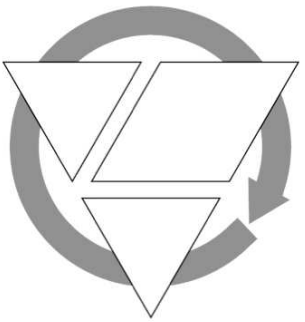

Figure 3.1: TMap model of essentials

3.1 Business driven explained

The key to testing is that tests are executed on the basis of test cases, checklists and the like. But what kind of tests are they? To ensure the tests' usefulness, they must be set up to test *those* characteristics and parts of a test object that represents a risk if it does not function adequately in production later on. This means that various considerations have already been made before test execution can begin. In other words, some thought has already been given to which parts of the test object need not be tested, and which must be tested and how and with what coverage. So what determines this? Why not test all parts of the test object as thoroughly as possible? If an organisation possessed unlimited resources, one option might indeed be to test everything as thoroughly as possible. But naturally, in real life an organisation rarely has the resources to actually do this, which means that choices must be made in what is tested and how thoroughly. Such choices depend on the risks that an organisation thinks it will incur, the available quantities of time and money, and the result the organisation wishes to achieve. The fact that the choices are based on risks, result, time and cost is called business driven and constitutes the basis for the BDTM approach. To understand and apply the BDTM approach, we first explain the concept of the "business case".

Business case as determining factor

IT projects must be approach increasingly from a purely economic perspective. The theory of IT governance controls projects on the basis of four aspects: result, risk, time and cost. For instance, it might be a more attractive investment for an organisation to start a high-risk project that potentially yields a high result than a project with very low risks where the benefits barely exceed the costs.

Normally, a business case is at the basis of an IT project. There are various definitions of business case, including the project-oriented one below according to [PRINCE2, 2002].

Definition

The business case provides the justification for the project and answers the questions: why do we do this project, which investments are needed, what does the client wish to achieve with the result?

During the project, the business case is verified at predefined points in time to ensure that the eventual results remain valid for the client. TMap supports the justification of IT, translating it to the activity of testing. TMap assumes that a project approach based on a business case complies with the following characteristics:

- The approach focuses on achieving a predefined result.
- The total project to achieve this result is realised within the available (lead) time.
- The project to achieve this result is realised at a cost in balance with the benefits the organisation hopes to achieve.
- The risks during commissioning are known and as small as possible. All of this within the framework set by the abovementioned characteristics.

The four IT governance aspects described above can be found in these characteristics.
For the successful execution of a project, it is important that the test process is aligned with the business case. The relationship between the business case and the test process is made via the business driven test management approach. In other words, with this approach, the business case characteristics can be 'translated' to the test process.

Characteristics of a business driven test management approach

Often test plans and reports fail to appeal to the client. The reason being that in the past the tester virtually always made decisions from an IT perspective. The test process was internally oriented and filled with test and IT jargon. This made it difficult to communicate with a non-IT client, such as a user department, even though this is extremely important.

TMap devotes explicit attention to communication due to the business driven test management approach.[1] BDTM starts from the principle that the selected test approach must enable the client to control the test process and (help) determine the test approach. This gives the testing an economic character. The required information to make this possible is delivered from the test process.

BDTM has the following specific properties:
- The total test effort is related to the risks of the system to be tested for the organisation. The deployment of people, resources and budget thereby focuses on those parts of the system that are most important to the organisation. In TMap, the test strategy in combination with the estimated effort is the instrument to divide the test effort over system parts. This provides insight into the extent to which risks are covered, or not.
- The estimated effort and planning for the test process are related to the

[1] Please note that BDTM is not an entirely accurate name. The word "business" suggests that it is intended exclusively for the link with the user departments, while testers clearly often still deal exclusively with IT departments. In this book, however, the general name BDTM is used.

defined test strategy. If changes are implemented that have an impact on the thoroughness of testing for the various system parts or systems, this is translated immediately to a change in the estimate and/or planning. The organisation thus is ensured of an adequate view of the required budget, lead time and relationship with the test strategy at all times.

- At various moments in the testing programme, the client is involved in making choices. The advantage is that the test process matches the wishes and requirements – and therefore the expectations – of the organisation as adequately as possible. Moreover, BDTM provides handholds to visualise the consequences of future and past choices explicitly.

The steps in the business driven test management approach

To understand the BDTM approach, it is important to keep an eye on the final objective. Which is to provide a quality assessment and risk recommendation about the system. Since not everything can ever be tested, a correct assessment can only be realised by dividing the test effort, in terms of time and money, as adequately as possible over parts and characteristics of the system to be tested. The steps of BDTM focus on this (see figure 3.2):

1. Formulating the assignment and gathering test goals.
   In consultation with the client, the test manager formulates the assignment, taking account of the four BDTM aspects: result, risk, time and cost.

   The test manager gathers the test objectives to determine the desired results of testing for the client. A test goal is a goal for testing relevant to the client and other acceptants, often formulated in terms of IT-supported business processes, realised user requirements or use cases, critical success factors, change proposals or defined risks (i.e., the risks to be covered).

2. Determining the risk class for each combination of characteristic and object part.
   When multiple test levels are involved, it is determined in a plan which test levels must be set up (master test plan). It is often already determined on the basis of a product risk analysis[2] what must be tested (object parts) and what must be investigated (characteristics).

[2] A product risk analysis (PRA) aims to ensure that the various stakeholders and test manager achieve a joint view of the more and less high-risk parts/characteristics of the system. The focus in the PRA is on the product risks, i.e. what is the risk to the organisation if the product does not have the expected quality?

If only one test level is involved, or if no or an overall product risk analysis was performed at the master test plan level, a (possibly supplementary) product risk analysis is performed within the relevant test level.

The eventual result (whether it is arrived at immediately or after one or more supplementary analyses) is a risk table defining a risk class related to the test goals and the relevant characteristic per object part ("Master test plan risk table").
A table then provides a guideline for the relative depth of testing per combination of characteristic/object part and test level ("Master test plan strategy table").

Now an *iterative process* emerges:

3. Determining whether a combination of characteristic and object part must be tested thoroughly or lightly.
   To determine the thoroughness of testing, the risk class per object part determined in the previous step is used as a starting point. Initially, the following applies: the greater the risk, the more thorough the required testing. The result is recorded in a strategy table per test level ("Test plan strategy table").

4. An overall estimate is then made for the test and a planning set up. This is communicated with the client and other stakeholders and, depending on their views, adjusted as necessary. In this case, steps 3 and 4 are executed once again. This emphatic gives the client control of the test process, enabling him to manage based on the balance between result and risk on the one hand and time and cost on the other.

End of iteration.

5. Allocating test techniques to the combinations of characteristic and object part.
   When the client and stakeholders agree on the estimate and the planning, the test manager completes a "Test design table". In here, the decisions concerning thorough and less thorough testing are translated to concrete statements about the targeted coverage. He then allocates test techniques to the combinations of characteristic and object part. The available test basis, among other things, is taken into account. These techniques are used to design and execute the test cases (and/or checklists) at a later stage. This is where the primary test process starts.

6. Throughout the test process, the test manager provides the client and other stakeholders with adequate insight into and control options over:
   - the progress of the test process

- the quality and risks of the test object
- the quality of the test process.

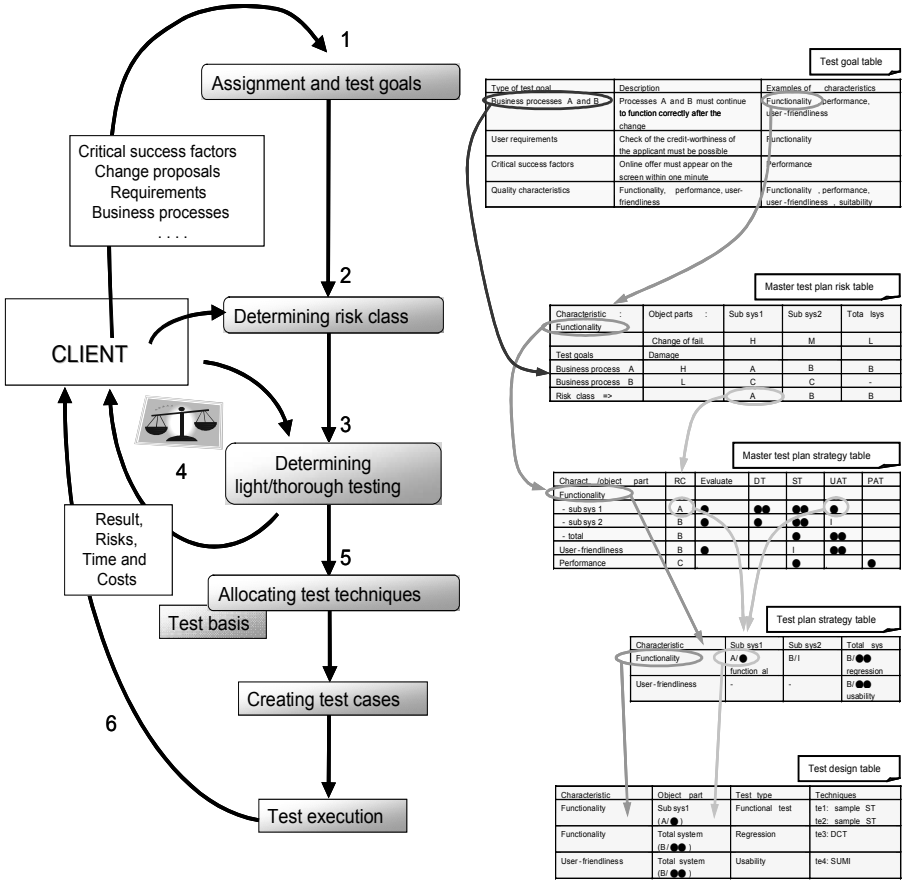

Figure 3.2: BDTM steps

In summary, the advantages of the BDTM approach are:

- The client having control over the process.
- The test manager communicates and reports in the terminology of the client with information that is useful in the client's context. E.g. by reporting in terms of test goals (such as business processes) instead of object parts and characteristics.
- At the master test plan level, detailing can be as intensive as required or possible. This may enable expending less effort on performing a product risk analysis or creating a test strategy for the separate test levels, or even to skip these steps (explanation of master test plan in subsequent section).

3.2 Structured test process

This section describes the phasing and activities in the following TMap processes:
- Master test plan, managing the total test process
- Acceptance and system tests
- Development tests.

Master test plan and other TMap processes

When the test manager, after consultation with the receiving parties, decides what will be tested for each test level, chances are that in the total picture of testing, certain matters will be tested twice unnecessarily. Or that certain aspects are ignored. The method should therefore be vice versa. A test manager, in consultation with the client and other stakeholders, makes a total overview of the distribution across test levels as to what must be tested when and with what thoroughness. The aim is to detect the most important defects as early and economically as possible. This agreement is defined in the so-called master test plan (MTP). This plan constitutes the basis for the test plans for the separate test levels. In addition to this content-based alignment, other types of alignment are: ensuring uniformity in processes (e.g. the defect procedure and testware management), availability and management of the test environment and tools, and optimal division of resources (both people and means) across the test levels.

This means that in addition to test levels like acceptance, system and development tests, the master test plan also plays an important part in TMap. Both for the master test plan and the test levels, it is important to set up a good process for creating plans and preparing, executing and managing activities.

While the goals of the acceptance and system tests differ, these test levels are not described separately, but as one single process. This was decided because the activities in both test levels are virtually the same and separate process descriptions would therefore have (too) much overlap.

In addition to these processes, the process “Supporting processes” has been defined because it is more efficient to organise certain aspects/support centrally than per project. This involves supporting processes for the following subjects:
- Test policy
- Permanent test organisation
- Test environments
- Test tools
- Test professional.

The supporting processes are discussed in relevant places as part of the complete tool box (see section 3.3).

3.2.1 Process: master test plan, managing the total test process

The master test plan provides insight into the various test and evaluation levels to be used, in such a way that the total test process is optimised. It is a management tool for the underlying test levels.

The process “Master test plan, managing the total test process” is split up into two phases: the Planning phase of the total test process and the Control phase of the total test process.

Planning phase of the total test process

The author of the MTP, often the test manager formulates the assignment, taking into account the four BDTM aspects of result, risks, time and cost, in consultation with the client. The test manager then works on the upcoming programme by having discussions with stakeholders and consulting information sources, such as documentation. In parallel, the test manager further elaborates the assignment and determines its scope in consultation with the client. In this phase, the first four steps of BDTM are executed: performing a PRA, establishing a test strategy, estimate and planning (see figure 3.2 “BDTM steps”).

Further activities in the creation of the plan are: the test manager defines the products that must be delivered by the test levels and makes a proposal as to the setup of the test organisation, centrally and overall per test level. The test manager aligns the infrastructure requirements of the test levels in order to deploy the – often scarce – test infrastructure as efficiently as possible. Test management can also be set up in part at the master test plan level. This can be achieved both by defining central procedures and standards for management and by the central management of certain aspects. Both options aim to prevent reinventing the wheel in the various test levels. The main risks threatening the test process are listed, and possible measures are proposed to manage these risks. As his last step, the test manager submits the master test plan to the client for approval.

Control phase of the total test process

The aim of this activity is controlling the test process, infrastructure and test products at the overall level to provide continuous insight into the progress and quality of the total test process and the quality of the test object.

Conformable to the frequency and form defined in the test plan, reports are made on the quality of the test object and the progress and quality of the test process. From the very first test activities, the testers develop a view of that quality. It is important that this is reported in every stage of the test process. The client receives periodical reports, and ad-hoc reports on request, on the condition of the system. Such reporting and adjustment are a vital part of the BDTM approach (BDTM step 6) and take place at both the level of the master test plan and that of the test level (figure 3.3 “Control and test processes”).

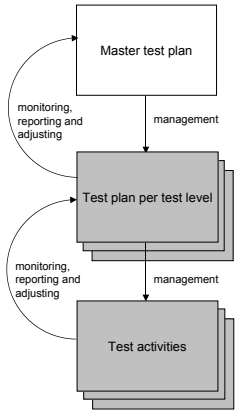

Figure 3.3: Control and test processes

3.2.2 Process: acceptance and system tests

The acceptance test and system test are considered as autonomous processes to be organised. They have their own test plan, their own budget, and often their own test environment to. They are processes running parallel to the development process, which must be started while the functional specifications are created. The TMap life cycle model is used both in the creation of the test plan and in the execution of the other activities in the test process.

Life cycle model

Like a system development process, a test process consists of a number of different activities. A test life cycle model is necessary to structure the various activities and their mutual order and dependencies. The life cycle model is a generic model. It can be applied to all test levels and test types and used in parallel with the life cycle models for system development. In the TMap life cycle model, the test activities are divided across seven phases: Planning, Control, Setting up and maintaining infrastructure, Preparation, Specification, Execution and Completion (see figure 3.4 “TMap life cycle model”). Each phase is split up into a number of activities.

Using a test life cycle model enables the organisation to keep an overview during the test process. By recording *what* has to be done *when, how, with what, where,* by *whom*, etc the claims to and the relationships with other aspects like techniques, infrastructure and organisation are made automatically.

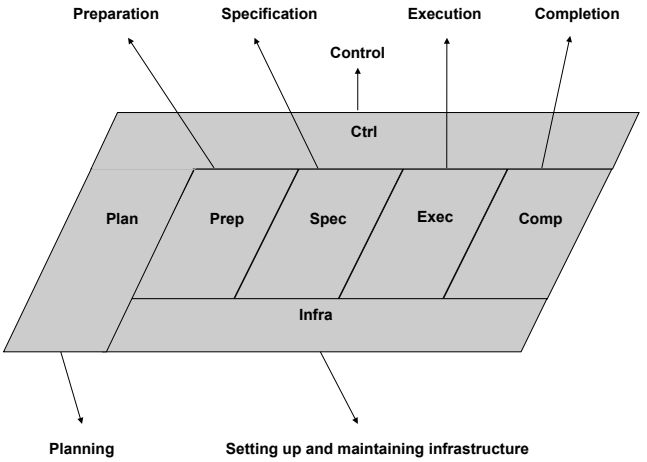

Figure 3.4: TMap life cycle model

The critical path and the shape of the life cycle model

If we were to compare the test process with an iceberg, only the Execution phase would be 'visible'. This means that only the Execution phase should be on the 'critical path' of a project. All activities in the other phases can be done either before or after.

The form of the life cycle model (parallelogram) shows that the test phases do not have to be executed strictly sequentially.

Test life cycle model relationships

The relationship between the TMap test life cycle and system development life cycle depends on the system development method used and the relevant test level. However, two 'fixed' relationships can be indicated. The start of the Preparation phase has a relationship with the moment at which the test basis becomes available; the start of the Execution phase has a relationship with the moment at which the test object becomes available.

Planning phase

The activities to be executed in the Planning phase create the basis for a manageable and high-quality test process. It is therefore important to start this phase as quickly as possible. The planning phase is an important test

phase but is almost always underestimated. Often, the framework for a certain test level is are already defined at the overall level in a master test plan. In this case, the detailed elaboration occurs in this phase.

After the test assignment has been finalised, an overall introduction to the test basis, subject matter and organisation (of the project) is made. It is impossible to test the system completely. Most organisations do not have the time and money for that. This is why the test strategy, estimate and planning are determined according to a risk analysis process (BDTM steps 1 through 4), of course always in consultation with the client. It is then determined which test techniques must be used (BDTM step 5). The objective is to realise the best achievable coverage at the right place within the defined BDTM frameworks. The first steps in setting up the test organisation and test infrastructure are also made. These activities are executed and laid down in the test plan for the relevant test level at the beginning of the test process.

Control phase

The primary test process is rarely executed according to plan. As such, the execution of the test plan also has to be monitored and adjusted, if necessary. This is done in the Control phase. The aim of the activities in this phase is to control and report on the test process in an optimal manner, such that the client has adequate insight into and control over the progress and quality of the test process and quality of the test object.

The test manager and/or administrator manage the test process, infrastructure and test products. Based on these data, the test manager analyses possible trends. He also ensures that he keeps well informed of the developments beyond testing, such as delays in development, upcoming big change proposals, and project adjustment. If necessary, the test manager proposes specific control measures to the client.

Information is the main product of testing. To this end, the test manager creates different kinds of reports for the various target groups, taking account of the BDTM aspects of result, risks, time and cost (BDTM step 6).

Setting up and maintaining infrastructure phase

The Setting up and maintaining infrastructure phase aims to care for the required test infrastructure and resources. A distinction is made between test environments, test tools and workplaces.

Setting up and maintaining the infrastructure represents a specific expertise. Testers generally have limited knowledge in this respect, but are highly dependent on it. No test can be executed without an infrastructure. All

responsibilities in relation to setting up and maintaining infrastructure are therefore usually assigned to a separate management department. In a testing programme, therefore, the team will have to collaborate closely with these other parties that may be external to the organisation. This means that test managers are in a situation in which they do not have control over the setup and maintenance of the infrastructure, but depend on it. This makes the setup and maintenance of the infrastructure an important area of concern for the test manager. It is a separate phase in the TMap life cycle model to maintain focus on it during the test. This phase runs in parallel to the Preparation, Specification, Execution and Completion phases. Dependencies with activities in other TMap test phases exist for some Setting up and maintaining infrastructure activities.

Preparation phase

The testability review of the test basis is done in the Preparation phase. The ultimate goal of this phase is to have access to a test basis of adequate quality to design the tests, which has been agreed with the client of the test.

Furthermore an early testability review of the test basis improves quality and prevents potential costly mistakes. This is because the development team works on developing the new information system on the basis of system documentation (which is part of the test basis). This documentation may contain errors that can cause a lot of – often costly – correction work if they are not detected in a timely manner. The earlier an error is found in a development process, the easier (and cheaper) it can be repaired.

Specification phase

The Specification phase specifies the required tests and starting situation(s). The aim is to prepare as much as possible so that tests can be executed as quickly as possible when the developers deliver the test object. This phase starts once the testability review of the test basis is completed successfully. The test specification runs in parallel to, and in the shadow of, the realisation of the software.

Execution phase

The aim of the Execution phase is to gain insight into the quality of the test object by executing the agreed tests.

The actual execution of the test starts when the test object, or a separately testable part of the test object, is delivered. The test object is first checked for completeness. It is then installed in the test environment to assess whether it functions as required. This is achieved by executing a first test, the so-called pretest. This is an overall test to examine whether the information system to

be tested, in combination with the test infrastructure, has sufficient quality for extensive testing. The central starting point is prepared if this is the case. The test can be executed on the basis of the test scripts created in the Specification phase. In this case, the starting point must be prepared for the test scripts that are to be executed. The test results are verified during execution. The differences between the predicted and actual results are registered, often in the form of a defects report.

Completion phase

The structured test approach of TMap can yield many benefits in the repeatability of the process. It allows products to be reused in subsequent tests if they comply with certain requirements. This may speed up certain activities. Products may be tangible things like test cases or test environments (testware), but also non-tangible things like experience (process evaluation).

When preserving the testware, a selection is made from the often large quantities of testware. Testware means, among other things, the test cases, test scripts and description of the test infrastructure. During the test process, an attempt was made to keep the test cases in line with the test basis and the developed system. If this was not (entirely) successful, the selected test cases are first updated in the Completion phase before the testware is preserved. The advantage of preserving testware this way is that it can be upgraded with limited effort when the system is changed to execute a (regression) test, for instance. There is therefore no need to design a completely new test.

The test process is also evaluated in this phase. The aim is to learn from the experiences gained and to apply these lessons learned in a new test, if any. It also serves as input for the final report, which the test manager creates in the Control phase.

3.2.3 Process: development tests

Development testing is understood to mean testing using knowledge of the technical implementation of the system. This starts with testing the first/smallest parts of the system: routines, units, programs, modules, objects, etc. After it has been established that the most elementary parts of the system are of acceptable quality, the larger parts of the system are subjected to integral testing. The emphasis here is on data throughput and the interfacing between e.g. the units up to the subsystem level.

Place of development tests

The development tests are an integral part of the development work executed by the developer. They are not organised as an autonomous process for an independent team. Despite that, a number of different activities for the development test process, with their mutual order and dependencies, can be identified and described with the aid of the TMap life cycle model. The detailed elaboration may vary per project or organisation and depends, among other things, on the development method used and the availability of certain quality measures.

An important quality measure is the concept of the agreed quality. To this end, the expectations of the client in relation to the craftsmanship and product quality must be made explicit during the planning to set up development testing. Examples of other quality measures are: test driven development, pair programming, code review, continuous integration, and the application integrator approach.

Differences between development and system/acceptance tests

The development test requires its 'own' approach that provides adequate elaboration of the differences between the development test and system/acceptance test as described below:

- Contrary to the system and acceptance tests, development tests cannot be organised as autonomous processes for more or less independent teams.
- Development testing uses knowledge of the technical implementation of the system, thereby detecting another type of defects than system and acceptance tests.
- In the development test, the person detecting the defects is often the same as the one who solves the defects.
- The perspective of development testing is that all detected defects are solved before the software is handed over.
- It is the first testing activity, which means that all defects are still in the product.
- Usually, the developers themselves execute development tests.

3.3 Complete tool box

TMap supports the correct execution of the structured test process with a complete tool box. The tool box focuses on working with the following subjects:
- Techniques: *how* it is tested
- Infrastructure: *where* and *with what* it is tested
- Organisation: *who* does the testing

The various tools are described in more detail in TMap at the moment they can be used. With the tool box, the tester possesses a great number of options to meet the test challenge successfully.

3.3.1 Techniques

Many techniques can be used in the test process. A test technique is a combination of actions to produce a test product in a universal manner.

TMap provides techniques for the following (see figure 3.5):
- Test estimation
- Defect management
- Creating metrics
- Product risk analysis
- Test design
- Product evaluation.

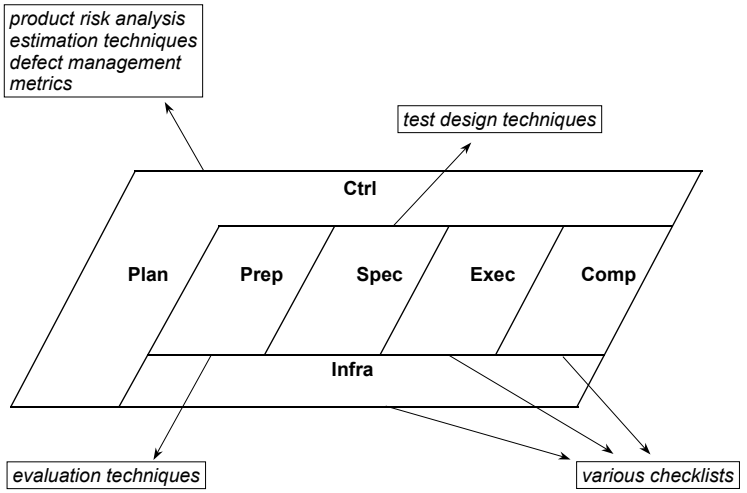

Figure 3.5: Test techniques

TMap also offers various checklists and overviews that can be used as a tool during the preparation and/or execution of certain activities.

The (groups of) test techniques are summarised in the following sections.

Test estimation

Estimates can be made at a number of different levels. The various estimation levels are shown in the figure below.

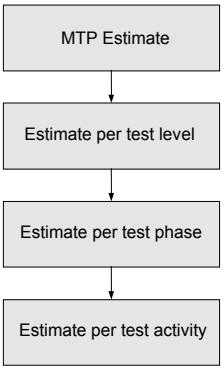

Figure 3.6: Estimation levels

Independent of the level, creating an estimate consists of the following generic steps:

1. Inventory the available material that can serve as a basis for the estimate.
2. Select (a number of) estimating techniques.
3. Determine the definitive estimate.
4. Present the outcome.

Choosing the estimating techniques in particular is a step requiring experience. You can select from several estimating techniques:

- Estimation based on ratios. Here, the test effort is generally measured against the development effort, e.g. in percentage ratios.
- Estimation based on test object size.
- Estimation using a 'Work Breakdown Structure'.
- Proportionate estimation based on the total test budget.
- Estimation on the basis of extrapolating experience figures from the beginning of the testing programme.
- Estimation on the basis of size and strategy using TMap's test point analysis (TPA).

Furthermore, TMap provides a technique to create an evaluation estimate.

Defect management

A defect is an observed difference between the expectation or prediction and the actual outcome. While the administration and monitoring of the defects is factually a project matter and not one of the testers, testers are usually very closely involved. A good administration must be able to monitor the lifecycle of a defect and provide various overviews. These overviews are used, among other things, to make well-founded quality statements.

Creating metrics

The definition, maintenance and use of metrics is important to the test process because it enables the test manager an answer, supported by facts, to questions like:

- What about the quality of the test object?
- What about the progress of the test process?

A structured approach to realise a set of test metrics is using the Goal-Question-Metric (GQM) method. In addition to describing the GQM method, TMap gives instructions to set up a practical test metrics starter set. It also provides a checklist that can be useful to make pronouncements on the quality of the object to be tested and the quality of the test process.

Product risk analysis

A product risk analysis (PRA) is analysing the product to be tested with the aim of achieving a shared view, among the test manager and other stakeholders, of the more or less risky characteristics and components of the product to be tested so that the thoroughness of testing can be agreed upon. The focus in PRA is on the product risks, i.e. what is the risk to the organisation if the product does not have the expected quality?

The result of the PRA constitutes the basis for the subsequent decisions in strategy as to light, thorough or non testing of a characteristic (e.g. a quality characteristic) or object part (component) of the product to be tested.

Test design

A test design technique is a standardised method to derive, from a specific test basis, test cases that realise a specific coverage. The implementation of test design techniques and their definition in the test specifications have several advantages:

- It provides a well-founded elaboration of the test strategy: the agreed coverage in the agreed place.
- It is a more effective way to detect defects than e.g. ad-hoc test cases.
- The tests are reproducible because the order and content of the test execution are described in detail.

- The standardised method ensures that the test process is independent of the individual who specifies and executes the test cases.
- The standardised method ensures that the test specifications are transferable and maintainable.
- It becomes easier to plan and manage the test process because the processes of test specification and execution can be split up into clearly definable blocks.

Test design techniques exist in many variants and combinations. The test design techniques described in TMap constitute a varied set with which most organisations can get to work immediately. TMap describes the following coverage types and test design techniques.

- Coverage types
  paths, decision points, equivalence classes, pair-wise testing, orthogonal arrays, limit value analysis, CRUD, operational and load profiles, right and fault paths, and checklists
- Test design techniques
  decision table test, data combination test, elementary comparison test, error guessing, exploratory testing, data cycle test, process cycle test, real-life test, semantic test, syntactic test, and use case test.

Product evaluation

TMap describes and uses the following evaluation techniques:

- Inspection
  In addition to determining whether the solution is adequately processed, an inspection focuses primarily on achieving consensus on the quality of a product.
- Review
  A review focuses primarily on finding courses for a solution on the basis of the knowledge and competencies of the reviewers, and on finding and correcting defects.
- Walkthrough
  A walkthrough is a method by which the author explains the contents of a product during a meeting. Several objectives are possible:
  - Bringing all participants to the same starting point, e.g. in preparation for a review or inspection process
  - Transfer of information, e.g. to developers and testers to help them in their programming and test design work, respectively
  - Asking the participants for additional information
  - Letting the participants choose from the alternatives proposed by the author.

Various checklists and overviews

TMap offers a great variety of checklists that will constitute a welcome addition to the tester when executing certain activities. For instance, there are checklists that can be used as support in taking stock of the assignment, determining the test facilities, determining the test project risks, establishing the test strategy, the evaluation of the test process, taking interviews, and determining whether adequate information is available to use a specific test design technique. TMap also offers other tools, such as an overview matrix of automated tools per TMap activity, a test type overview, and criteria to select a tool.

These tools and many more can be found on and downloaded from www.tmap.net.

3.3.2 Infrastructure

Test environments, test tools and workplaces are necessary to execute tests.

Test environments

A fitting test environment is necessary for dynamic testing of a test object (running software). A test environment is a system of components, such as hardware and software, interfaces, environmental data, management tools and processes, in which a test is executed. The degree to which it can be established in how far the test object complies with the requirements determines whether a test environment is successful. The setup and composition of a test environment therefore depend on the objective of the test. However, a series of generic requirements with which a test environment must comply to guarantee reliable test execution can be formulated. In addition to being representative, manageable and flexible, it must also guarantee the continuity of test execution.

Setting up and managing the test environment represents an expertise of which testers generally have no knowledge. This is why a separate department – outside the project – is generally responsible for setting up and managing the test environment.

Test tools

To execute the tests efficiently, tools in the form of test tools are necessary. A test tool is an automated instrument that provides support to one or more test activities, such as planning and control, test specification, and test execution. The use of tools can have the following advantages:

- Increased productivity
- Higher testing quality
- Increased work enjoyment
- Extension of test options.

The test tools are classified in four groups:
- Tools for planning and managing the test
- Tools for designing the test
- Tools for executing the test
- Tools for debugging and analysing the code.

Workplaces

One of the aspects that is often forgotten in testing, is the availability of a workplace where testers can do their job under good conditions, effectively and efficiently. This means office setup in the broadest sense since the testers must also be able to do their work under good conditions. The workplace is therefore more than just office space and a PC. Matters such as access passes, power supply and facilities to have lunch must be arranged. At first sight, the workplace for a tester does not differ much from the regular workplace. But appearances can be deceptive. What is tested is often new to the organisation and the workplace. Testers may have to deal with the situation that their workplace is not yet prepared for the new software. For example, testers often require separate authorisations. They must, for instance, be able to install the new software on their local PC. This may also be necessary for the use of certain test tools.

3.3.3 Organisation

Test processes that are not adequately organised usually have disastrous results. The involvement of many different disciplines (see figure 3.7), conflicting interests, unpredictability, complex management tasks, lack of experience (figures), and time pressure make setting up and managing the test organisation a complex task. On the one hand there is the organisation in the test team where everyone must have their tasks and responsibilities. On the other, the test team must be an integral part of the project organisation.

A test organisation can be seen as the creation of effective relationships between test functions, test facilities and test activities to issue good quality advice in time.

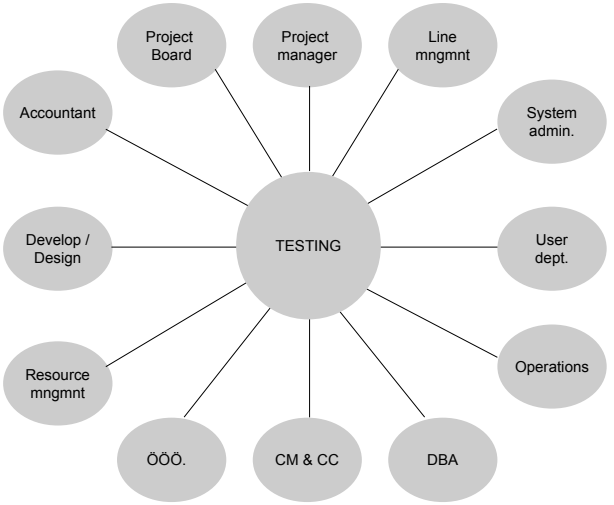

Figure 3.7: The many disciplines involved

The organisation of structured testing requires attention for the following fields:

- Test policy
- Permanent test organisation
- Test organisation in projects
- Test professional
- Test roles.

Test policy

The test policy describes how an organisation deals with the people, resources and methods involved in the test process in the various situations. Since testing is one of the tools to ensure quality, the test policy will have to be in line with the other policy measures and initiatives in relation to quality management. We recommend making sure that the test policy is in line with the strategic, tactical and operational policy of the organisation.

Permanent test organisation

A permanent test organisation, contrary to project-based testing, does not elaborate a specific element of the test process on a per project basis, but across all projects. Reasons to create such an organisation are, among other things, the improved leverage of scarce expertise, standardisation of test products, limiting the test project start-up time, continuous improvement of the test process, consolidation of experiences, and prior insight into the test costs and lead time.

Test organisation in projects

At the start of a test project, the roles, tasks, authorisations and responsibilities for the test project are defined. This can be done for the total test process (i.e. across all test levels), or for one specific test level. The relationship between the specified roles, the separate test levels, and the relationships with the other stakeholders in the system development process are then determined and laid down. The test manager must not forget to establish the relationship with a test or quality department, if any.

How this is all set up specifically depends heavily on the organisation type selected for the test work. The choice depends on the test level, project and organisation. The test manager can sometimes – but not always – influence this decision. Roughly, the following organisation types can be distinguished:

- Testing as an autonomous activity in a project
- Testing integrated in a project
- Testing as an autonomous line organisation
- Testing integrated in a line organisation.

Test professional

A great variety of expertise is required for a tester to be able to function well in the discipline of testing. A tester must have knowledge in the field of the subject matter, the infrastructure (test environment, development platform, test tools) and testing itself. What are a tester's characteristics, in other words, what properties must a person have to be an ideal tester? While many can be listed, the tester must at least:

- have verbal and written communication skills
- be able to work accurately and have analytical skills
- be convincing and persevering
- be factual and have a positively critical attitude
- be creative.

Test roles

The execution of test activities in a project or in the line requires that the tasks are defined and that the executor of the tasks has the right knowledge and competencies.

Roles and positions can be distinguished in this respect. A role is the description of one or more tasks with the required knowledge and competencies. There are roles that match positions one-on-one. There are also roles that do not exist as a position.

Differences and similarities between roles and positions:
- A role aims at fulfilling tasks for the test project or permanent test organisation.
- A position focuses on the employee and his place in the career cube.
- They share the tasks to be executed and the required knowledge competencies.

3.4 Adaptive and complete method

TMap is an approach that can be applied in all test situations and in combination with any system development method. It offers the tester a range of elements for his test, such as test design techniques, test infrastructure, test strategy, phasing, test organisation, test tools, etc. Depending on the situation, the tester selects the TMap elements that he will deploy. There are situations in which only a limited number of elements need to be used; but in others he will have to use the whole range of elements. This makes TMap an adaptive method, which in this context is defined as:

Definition

Adaptive is the ability to split up an element into sub-elements that, in a different combination, result in a new, valuable element for the specific situation.

The adaptiveness of TMap is not focused on a specific aspect of the method, but is embedded throughout the method. Adaptiveness is more than just being able to respond to the changing environment. It is also being able to leverage the change to the benefit of testing. This means that TMap can be used in every situation *and* that TMap can be used in a changing situation. In the course of projects and testing, changes may occur that have an impact on earlier agreements. TMap offers the elements to deal with such changes.

TMap's adaptiveness can be summarised in four adaptiveness properties:
- *Respond* to changes
- *(Re)use* products and processes
- *Learn* from experiences
- *Try* before use

These properties are explained in further detail below.

3.4.1 Respond to changes

Adaptiveness starts with determining the changes and responding to them. In TMap, this happens from the very beginning in the earliest activities of the (master) test plan. When determining and taking stock of the assignment, obtaining insight into the environment in which the test is executed and establishing possible changes play a major part. This is precisely where the basis is created for the further elaboration and implementation of the method. Which test levels, test types, phases, and tools are used and how? But it is not limited to these activities. The test strategy and associated planning are defined in close consultation with the client. If the test strategy and derived estimate and planning are not acceptable to the client, the plan is adapted. This emphatic gives the client control of the test process, enabling him to manage based on the balance between result and risk on the one hand and time and cost on the other. Such feedback is provided throughout the testing programme, and in the control phase, the test manager may also decide to adapt certain aspects of the test plan in consultation with the client.

3.4.2 (Re)use products and processes

Being able to use products and processes quickly is a requirement for adaptiveness. TMap offers this possibility, among other things thanks to the large quantity of tools included in the form of test design techniques, checklists, templates, etc. These can be found in the book and on www.tmap.net.
In addition to use, reuse plays an important part. The emphasis in this respect lies in the Completion phase, where the activities are defined to identify what can be reused and how it can be optimally preserved. TMap offers various forms of a permanent test organisation for the organisational anchoring of the reuse of products and processes.

3.4.3 Learn from experience

As a method, TMap offers the space to learn and apply what was used. Therefore the activity evaluating the test process is incorporated into the test process. An other important instrument is the use of metrics. For the test process, metrics on the quality of the test object and the progress and quality of the test process are extremely important. They are used to manage the test process, justify the test recommendations, and compare systems or test processes. Metrics are also important to improve the test process through assessing the consequences of certain improvement measures.

3.4.4 Try before use

TMap offers room to try something before it is actually used. The main instruments here are the activities relating to the intake. The intake of the test basis (using a testability review), of the test infrastructure, and of the test object allow one to try first before actually using.

Implementing TMap does not mean that everything in the book should be used without question. Another form of trying before using is therefore 'customising' TMap to fit a specific situation. A selection can be made from all of the TMap elements to achieve this. After the approach, customised to the situation, has been tried out ('pilot'), it can be rolled out in the organisation.

For many situations, 'customising' TMap has already been done. The specific TMap approach for a certain situation (known under the name 'expansion') can be found on www.tmap.net and in the TMap Test Topics book [Koomen, 2005].

Glossary

Acceptance test
A test executed by the user(s) and manager(s) in an environment simulating the operational environment to the greatest possible extent, that should demonstrate that the developed system meets the functional and quality requirements.

Adaptive
The ability to split up an element into sub-elements that, in a different combination, result in a new, valuable element for the specific situation.

Basic technique
The method of deriving test situations from the test basis to achieve the required coverage type.

BDTM
Business driven test management is aimed at enabling the client to manage the test process on rational and economic grounds. Important BDTM aspects are: result, risk, time and cost.

Boundary value analysis
Test principle based on the fact that a test around a boundary has a greater chance to detect a defect.

Business case
The business case provides the justification for the project and answers the questions: why do we do this project, which investments are needed, what does the client wish to achieve with the result?

Central starting point
See Starting point.

Chain test
See End-to-end test.

Checklist (coverage type)
All the situations are tested that are summed up in an unstructured list.

Code review
A method of improving the quality of written code by evaluating the work against the specifications and/or guidelines and subjecting it to peer review.

Combined test
Test approach by which the system test and the functional acceptance test are combined to a single test level.

Completeness
The certainty that *all* inputs and changes are processed by the system.

Condition coverage
The possible outcomes of ("true" or "false") for each condition are tested at least once.

Condition/decision coverage
The possible outcomes of each condition and of the decision are tested at least once. This implies both "condition coverage" and "decision coverage".

Connectivity
How easy a link with a different information system or within the information system can be made and modified.

Continuity
The certainty that the information system will continue uninterruptedly, which means that it can be resumed within a reasonable time even after serious interruptions.

Correctness
The degree to which the system processes the input and changes entered correctly, in accordance with the specifications, to produce consistent data sets.

Coverage
Coverage is the ratio between that which can be tested and that which is tested with the test set.

Coverage ratio
The percentage of test situations, as defined by the coverage type, that is covered by the test.

Coverage type
The form in which the covering of test situations deducible from the test basis, is expressed.

Data controllability
The ease with which the correctness and completeness of the information (in the course of time) can be checked.

Decision coverage
The possible outcomes of the decision are tested at least once.

Decision point
A combination of one or more conditions that define the conditions for the various possibilities in the subsequent system behaviour.

Defect (fault)
The result of an error residing in the code or document.

Degradation factor
The ease with which the core of the information system can continue after a part has failed.

Driver
A simulation program that replaces a program that should take care of the control and/or the calling of the test object.

Dynamic testing
Testing by execution of the test object and/or the running of software.

Effectivity
The degree to which the information system meets the demands of the organisation and the profile of the end users for whom it is intended, as well as the degree to which the information system contributes to the achievement of business objectives.

Efficiency
The relationship between the performance level of the system (expressed in the transaction volume and overall speed) and the amount of resources (CPU cycles, I/O time, memory and network capacity, etc.) that are used.

End-to-end test
A test type where the end-to-end functionality of one or more systems is tested with end-to-end test cases.

Equivalence class
In the application of equivalence classes, the entire value range of a parameter is partitioned into classes, in which the system behaviour is similar (equivalent).

Error
Human mistake; this action takes place prior to any faults and/or failures.

Evaluation
Evaluation is assessing the intermediary products in the system development process.

Exploration Testing
Is the simultaneous learning, designing and executing of tests, in other words every form of testing in which the tester designs his tests during test execution and the information obtained is reused to design new and improved test cases.

Fail-over possibilities
The ease with which (part of) the information system can continue elsewhere.

Failure
The result or manifestation of one or more faults. When the system is performing differently from the required behaviour, from a viewpoint outside the system. Users will see the failure.

Fault (defect)
The result of an error residing in the code or document. Fault is the view from inside the system. Fault is the state where mistake or error exists. Developers will see the fault.

Flexibility
The degree to which the user may introduce extensions or modifications to the information system without changing the program itself. Or: the degree to which the system can be modified by the controlling organisation without being dependent on the IT department for maintenance.

FPA functions
Subdivision of user functions in FPA functions: logical set of data, links, input functions, output functions, reading functions. These FPA functions are the elementary building blocks to determine the functionality of a system.

Function point
Unit to measure the functionality and/or the size of application software.

Function point analysis (FPA)
A method aiming to measure the size of the functionality of an automated system. The measurement is independent of the technology. This measurement may be used as a base for the measurement of productivity, the estimation of the needed resources, and project control.

Functional acceptance test
A test carried out by the future user(s) in an optimally simulated production environment, with the aim of demonstrating that the developed system meets the functional requirements.

Functionality
The certainty that data processing is correct and complete, in accordance with the description in the functional specifications.

Generic Test Agreements
The overall approach for the setup and organisation of test processes that applies to more than one project or release. General agreements on e.g. the test process, standard strategy, estimating method, procedures, organisation, communication, documentation, etc.

Infrastructure (suitability of)
The suitability of hardware, network, systems software and DBMS for the application concerned and the degree to which the elements of this infrastructure interrelate.

Initial situation
Everything that is needed to prepare the system for receiving the required input. This includes not only the data that are needed for the processing, but also the condition in which the system and its environment must be. For instance, one might think of setting a specific system date, or running specific week and month batches that bring the system to a specific status.

Inspection
A formal evaluation technique, with products being read thoroughly by a group of experts. In addition to determining whether the solution is adequately processed, an inspection focuses primarily on achieving consensus on the quality of a product. The aim of the inspection is to help the author find as many deviations as possible in the available time.

Known errors
Defects that have been found but have not been solved (yet).

Logical test case
Describes, *in logical terms*, the circumstances in which the system behaviour is examined by indicating which *test situations* are covered by the test case.

Maintainability
The ease with which the information system can be adapted to new demands from the user, to changing external environments, or in order to correct defects.

Manageability
The effort needed to get and keep the information system in its operational state.

Master test plan
Test plan by which the various test levels are geared to one another.

Modified condition/decision coverage
Every possible outcome of a condition is the determinant of the outcome of the decision, at least once.

Multiple condition coverage
All the possible combinations of outcomes of conditions in a decision (therefore the complete decision table) are tested at least once. This implies "modified condition/decision coverage".

Object part
A logically coherent part of the test object from the perspective of the characteristic to be tested.

Online
Function mode of an information system in which the information system immediately processes the commands and directly shows the answer (output) on the screen (or otherwise).

Orthogonal array
An orthogonal array $L_N(s^k, t)$ is a 2-dimensional array of N rows and k columns consisting of elements that can assume s values, whereby every combination of t columns contains all the combinations of the s values in equal proportion.

Pairwise testing
Tests all the possibilities of any combination of 2 factors.

Performance
The speed with which the information system processes interactive and batch transactions.

Permanent test organisation
A line organisation that offers test services.

Physical test case
The *concrete elaboration* of a logical test case, with choices having been made for the values of all required the input and settings of the environmental factors.

Portability
The diversity of the hardware and software platforms on which the information system can run, and how easy it is to transfer the system from one environment to another.

Pre-test
Testing the delivered product in such a way that it is determined whether or not the product is of sufficient quality to execute a complete test of this product.

Product risk
The chance that the product fails in relation to the expected damage if this occurs:
Product risk = Chance of failure * Damage
where Chance of failure = Chance of defects * Frequency of use

Product risk analysis
Analysing the product to be tested with the aim of achieving a joint view, for the test manager and other stakeholders, of the more or less risky characteristics and parts of the product to be tested so that the thoroughness of testing can be related to this view.

Production acceptance test
A test carried out by the future administrator(s) in an optimally simulated production environment, with the aim of demonstrating that the developed system meets the requirements set by system management.

Quality
The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs.

Quality assurance
All the planned and systematic activities necessary to provide adequate confidence that a product or service meets the requirements for quality.

Quality characteristic
A quality characteristic describes a property of an information system.

Recoverability
The ease and speed with which the information system can be restored after an interruption.

Regression
Regression is the phenomenon that the quality of a system deteriorates as a whole as a result of individual amendments.

Regression test
A regression test is aimed at verifying that all the unchanged parts of a system still function correctly after the implementation of a change.

Reliability
The degree to which the information system remains free from interruptions.

Reusability
The degree to which parts of the information system, or the design, can be reused for the development of different applications.

Review
An evaluation technique where a product (60-80% complete) is submitted to a number of reviewers with the question to assess it from a certain perspective (depending on the review type). A review focuses primarily on finding courses for a solution on the basis of the knowledge and competencies of the reviewers, and on finding and correcting defects. There are various review types, such as: technical review (e.g. selecting solution direction/alternative), management review (e.g. determining project status), peer review (review by colleagues), and expert review (review by experts).

Risk reporting
A description of the extent to which the system meets the specified quality requirements and the risks associated with bringing a particular version into production, including any available alternatives.

Robustness
The degree to which the information system proceeds as usual even after an interruption.

Role
Describes one or more tasks and the knowledge and skills required to carry them out.

Security
The certainty that data can be viewed and changed only by those who are authorized to do so.

Starting point
Initial situations often contain the same data for several test cases. Such data are therefore included in a so-called starting point for the entire test and not separated for each test case. It is called a central starting point if this is intended for more tests or testers.

Static testing
Testing by examining products (such as manuals or source code) without any programs being executed.

Stub
A simulation program.

Suitability
The degree to which manual procedures match the automated information system and the fitness for use of these manual procedures for the organisation.

System integration test
A test carried out by the future user(s) in an optimally simulated production environment, with the aim of demonstrating that (sub)system interface agreements have been met, correctly interpreted and correctly implemented.

System management
System management is responsible for technical operation of the software in its intended infrastructure in production.

System test
A test carried out by the supplier in a (manageable) laboratory environment, with the aim of demonstrating that the developed system, or parts of it, meet with the functional and non-functional specifications and the technical design.

Test basis
The test basis is the information that defines the required system behaviour.

Test case
Used to examine whether the system displays the desired behaviour under specific circumstances.

Test depth level
Test depth level N = the certainty that all the combinations of N consecutive paths are covered.

Test design technique
A standardised method of deriving test cases from a particular test basis that will achieve a certain coverage.

Test environment
A composition of parts, such as hardware and software, connections, environment data, maintenance tools and management processes in which a test is carried out.

Test goal
A goal that, for the client, is relevant for testing, often formulated in terms of business processes supported by IT, realised user requirements or use cases, critical success factors, change proposals or cited risks to be covered.

Test harness
A collection of software and test data configured for a development environment with the purpose of dynamically testing one unit or a series of units, whereby the behaviour and output are checked.

Test infrastructure
Consists of the facilities and resources necessary to facilitate the satisfactory execution of the test. A distinction is made between test environments, test tools and workplaces.

Test level
A test level is a group of test activities that are managed and executed collectively.

Test line
The operational organisation to provide test services to one or more clients. A test line has a fixed team of testers, infrastructure, test tools and standardised work procedures.

Test object
The test object is the information system (or part thereof) to be tested.

Test organisation
The whole of the test functions, facilities, procedures and activities including their relationships.

Test pattern
A general solution for a specific recurring test problem.

Test plan
In a test plan the general structure and the strategic choices with respect to the test to be executed are formulated. The test plan forms the scope of reference during the execution of the test and also serves as an instrument to communicate with the client of the test. The test plan is a description of the test project, including a description of the activities and the planning; therefore it is *not* a description of the tests themselves.

Test point
Unit of measurement for the size of the high-level test to be executed.

Test point analysis (TPA)
A method with the possibility to perform a technology-independent measurement of the test depth level of an information system, on the basis of a function point analysis, and to use this measurement as a basis for a productivity measurement, an estimate of the required resources, and project management.

Test policy
Describes how an organisation deals with the people, resources and methods involved with the test process in the various situations.

Test process
The collection of tools, techniques and working methods used to perform a test.

Test script
Combines multiple physical test cases to be able to execute them in an efficient and simple manner.

Test situation
An isolated condition under which the test object displays a specific behaviour that needs to be tested.

Test strategy
The distribution of the test effort and coverage over the parts to be tested or aspects of the test object aimed at finding the most important defects as early and cheaply as possible.

Test team
A group of people who, led by a test manager, undertake test activities.

Test technique
A set of actions aimed at creating a test deliverable by a universal method.

Test tool
An automated instrument that supports one or more test activities, such as planning, control, specification and execution.

Test tool policy
Describes how an organisation handles the acquisition, implementation and use of test tools in the various situations.

Test type
A group of test activities with the intention of checking the information system in respect of a number of correlated (part aspects of) quality characteristics.

Test unit
A collection of processes, transactions and/or functions that are tested collectively.

Testability
The ease and speed with which characteristics of the system can be tested (following each adjustment).

Testability review
The detailed check of the test basis on testability.

Testing
Testing is a process that provides insight into, and advice on, quality and the related risks.

Testing (ISO)
Technical operation that consists of the determination of one or more characteristics of a given product, process or service according to a specified procedure [ISO/IEC Guide 2, 1991].

Testware
All the test documentation produced in the course of the test process that can be used for maintenance purposes and that should therefore be transferable and maintainable.

Unit integration test
A test carried out by the developer in the development environment, with the aim of demonstrating that a logical group of units meets the requirements defined in the technical specifications.

Unit test
A test carried out in the development environment by the developer, with

the aim of demonstrating that a unit meets the requirements defined in the technical specifications.

User function
A property recognized by the user which the delivered product should meet. Generally speaking the user functions may best be described as objects and processes.

User-friendliness
How easy it is for end users to use the system. This general definition is often divided into how easy it is for end users to learn to work with the information system, and how easy it is to work with for trained users.

Users acceptance test
A test carried out by the future user(s) in an optimally simulated production environment, with the aim of demonstrating that the developed system meets the requirements of the users.

Walkthrough
An evaluation technique by which the author explains the contents of a product during a meeting. Several different objectives are possible: bringing all participants to the same starting point, transfer of information, asking the participants for additional information or letting the participants choose from the alternatives proposed by the author.

References

- [Aalst, 1999]
  Aalst, L. van der, Koning, C. de, *Testing expensive? Not testing is more expensive!,* Informatie, October 1999, www.tmap.net
- [Abran, 2003]
  Abran, A. (2003), *Implementation Guide for ISO/IEC 19761:2003 (Measurement Manual)*, version 2.2, Common Software Measurement International Consortium, www.cosmicon.com
- [Bach, 2000]
  Bach, Jonathan, (2000) Session-based test management, www.satisfice.com
- [Bach, 2002]
  Bach, James, (2002) Exploratory testing explained, www.satisfice.com
- [Bach, 2003]
  Bach, J. (1996-2002), From his workshop Rapid Software Testing
- [Basili, 1994]
  Basili, V.R., Galdiera, G. and Rombach, H.D. (1994), *The Goal-Question-Metric Approach*, John Wiley & Sons, Inc., New York
- [Basili, 1997]
  Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sorumgard, S., Zelkowitz, M., *The Empirical Investigation of Perspective-Based Reading*, also appeared in: Empirical Software Engineering, 2, 1997
- [Beck, 2002]
  Beck, Kent, Test Driven Development By Example (2002) Addison-Wesley Professional, ISBN 0321146530
- [Beizer, 1990]
  Beizer, B. (1990), *Software Testing Techniques*, International Thomson Computer Press.
- [Belbin, 2003]
  Belbin, R Meredith, Management Teams – Why They Succeed or Fail (second edition) (2003), Butterworth Heinemann, ISBN: 0 7506 5910 6
- [Bieberstein, 2005]
  Bieberstein, Norbert; Bose, Sanjay; Fiammante, Marc; Jones, Keith and Shah, Rawn, *Service-Oriented Architecture (SOA) Compass, Business Value, Planning and Enterprise Roadmap*, (2005) IBM Press, ISBN 0-13-187002-5
- [Black, 2002]
  Rex Black, keynote at Eurostar2002, Kopenhagen
- [Black, 2004]
  Black, Rex, *Critical Testing Processes* (2004) Addison Wesley, ISBN 0-201-74868-1

- [Boehm, 1981]
  Boehm, B.W. (1981), *Software Engineering Economics*, Prentice-Hall Inc., Englewood Cliffs, ISBN 0-13-822122-7
- [Broekman, 2003]
  Broekman, B., Notenboom, E. (2003), *Testing Embedded Software*, Addison-Wesley, London, ISBN 0-321-15986-1
- [Brooks, 1975/1995]
  Brooks, Frederic P., The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition, (1975/1995), Addison-Wesley Professional, ISBN 0201835959
- [Clifton, 2004]
  Clifton, Marc, Advanced Unit Testing (2004), http://www.codeproject.com/gen/design/autp5.asp
- [Cockburn, 2000]
  Cockburn, Alistair and Williams, Laurie, The Costs and Benefits of Pair Programming, Humans and Technology Technical Report 2000.01, Jan 2000, http://alistair.cockburn.us
- [Collard, 1999]
  Collard, R. (1999), *Test Estimation*, STAREast 1999
- [Copeland, 2003]
  Copeland, L. (2003), *A Practitioner's Guide to Software Test Design*, Artech House Publishers, ISBN 1-58053-791-X
- [Deming, 1992]
  Deming, W. Edwards (1992), *Out of the crisis*, University of Cambridge, ISBN 0-521-30553-5
- [Fagan, 1986]
  Fagan, M.E. (1986), *Advances in Software Inspections*, in: IEEE Transactions on Software Engineering, July 1986
- [Fewster, 1999]
  Fewster, M., Graham, D. (1999), *Software Test Automation*, Addison/Wesley US, ISBN 0-201-331403
- [Gilb, 1993]
  Gilb, T. & R. Graham (1993), *Software Inspection*, Addison Wesley, ISBN 0-201-63181-4
- [Hedayat, 1999]
  Hedayat, A.S., Sloane, N.J.A. and John Stufken (1999) *Orthogonal arrays: Theory and Applications*, Springer Verlag, ISBN 0-387-98766-5
- [IEEE, 1998]
  The Institute of Electrical and Electronics Engineers, Inc. (1998), *IEEE Std 1028-1997 Standard for Software Reviews*, 345 East 47th Street, New York, NY 10017-2394, USA, ISBN 1-55937-987-1
- [IFPUG, 1994]
  IFPUG (International Function Point User Group) (1994), *Function Point Counting Practices*, release 4.0, IFPUG, January 1994

- [ISO/IEC, 1991]
  ISO/IEC Guide 2 (1991), *General terms and definitions concerning standardization and related activities*, International Organization of Standardization
- [ISO 9126-1, 1999]
  ISO/IEC 9126 part 1 (1999), *Information Technology - Software Product Quality - Part 1: Quality Model*, International Organization of Standardization
- [ISO, 1994]
  ISO 8402 (1994), *Quality Management and quality assurance - vocabulary*, International Organization of Standardization
- [Jones, 1996]
  Jones, Capers, (1996), *Applied Software Measurement: Assuring Productivity & Quality*, McGraw Hill Text, ISBN 0070328269
- [Kaner, 1999]
  Kaner, C., Falk, J., Nguyen, H.Q. (1999), Testing computer software, 2nd edition, Wiley, ISBN 0-471-35846-0
- [Kaner, 2001]
  Kaner C., Bach J. (2001), *Exploratory testing in pairs*, StarEast 2001
- [Koomen, 1999]
  Koomen, T., Pol, M. (1999), *Test Process Improvement: a practical step-by-step guide to structures testing, Addison-Wesley London*, ISBN 0 210 59624 5
- [Koomen, 2005]
  Koomen, T., Baarda, R., (2005), *TMap® Test Topics* (part I), Tutein Nolthenius, 's-Hertogenbosch, ISBN 90-72194-75-6
- [Laitenberger, 1995]
  Laitenberger, O., (1995), *Perspective based Reading: Technique, Validation and Research in Future*, Student project (Projektarbeit), Department of Computer Science, University of Kaiserslautern, 67653 Kaiserslautern, Germany
- [Lyndsay, 2002]
  Lyndsay, James (2002), Adventures in session-based testing, Proceedings EuroSTAR 2002
- [McCabe, 1976]
  McCabe, T.J. (1976*), A complexity metric*, IEEE Transactions on software engineering, vol. 2, IEEE Press
- [McCall, 1977]
  McCall, J.A., P.K. Richards en G.F. Walters (1977), *Factors in software quality*, RADC-TR-77-363 Rome Air Development Center, Griffis Air Force, Rome (New York, USA)
- [Musa, 1998]
  Musa, J.D. (1998), *Software Reliability Engineering*, McGraw-Hill
- [Nielsen, 1999]
  Jakob Nielsen, Designing Web Usability: The Practice of Simplicity, (1999) New Riders Publishing, Indianapolis, ISBN 1-56205-810-X Nielsen Jakob, www.useit.com, 2006
- [Pettichord, 2000]

Pettichord, Bret, Testers and Developers Think Differently; Understanding and utilizing the diverse traits of key players on your team, Jan/Feb 2000, STQE Magazine

- [PRINCE2, 2002]
  Managing Successful Projects with PRINCE2 (2002), The Stationary Office London, ISBN 0-11-330891-4
- [Roden, 2005]
  Lloyd Roden, Choosing and Managing the Ideal Test Team, winter 2005 issue, www.methodsandtools.com
- [Rothman, 2006]
  Johanna Rothman, "Hiring The Best Knowledge Workers, Techies & Nerds: The Secrets & Science Of Hiring Technical People, (2006) Dorset House Publishing Company, ISBN 0932633595
- [Schaefer, 1996]
  Schaefer, H., (1996), *Surviving under time and budget pressure*, in Proceedings EuroSTAR Conferencce 1996, Amsterdam. Also home.c2i.net/schaefer/testing/risktest.doc
- [Splaine, 2002]
  Splaine, Steven, Testing Web Security: Assessing the Security of Web Sites and Applications (2002) Wiley, ISBN 0471232815
- [The Standish Group, 2003]
  The Standish Group, (2003), *What Are Your Requirements?*, The Standish Group International, West Yarmouth
- [Test Cube, 2006]
  The Test Cube, Expertise group Sogeti, 1.0, October 2006, www.tmap.net
- [Tufte, 2001]
  Edward Tufte, The visual display of quantitative information, 2nd edition, 2001, Graphics Press LLC, ISBN 0961392142
- [Vaaraniemi, 2003]
  Vaaraniemi, Sami, The benefits of automated unit testing, November 2003, http://www.codeproject.com
- [Whittaker, 2000]
  Whittaker, J., Jorgenson, A. (2000), *How to break software*, proceedings Eurostar 2000
- [Whittaker, 2003]
  Whittaker, James A., Thompson, Herbert H., *How to Break Software Security* (2003) Addison Wesley, ISBN 0321194330
- [www.w3c.org]
  Web Content Accessibility Guidelines 1.0 en 2.0

About Sogeti

Sogeti forms the Capgemini's Local Professional Services Division, counting over 15,000 employees. Sogeti offers a package of local services for large enterprises.
Designing, developing, implementing, testing and managing IT solutions is our core-business. One of our specializations is our testing and QA offering, Software Control Testing (SCT). Proof of this are our internationally recognized methods TMap®, the Test Management approach for structured software testing and TPI®, the Test Process Improvement model for improving test processes.

Software Control Testing

We understand the importance of getting maximum business value from your software, especially in today's challenging and increasingly competitive economic environment. Currently, organizations are under intense pressure to develop high-quality software more quickly, making the risk associated with insufficient software quality greater. Effective software testing is one of the most powerful actions that an organization can take to control and minimize these risks.
Built on practical "real-world" experience, Sogeti's SCT solution takes a proactive and structured approach to software testing. We help organizations detect and correct defects early in the development lifecycle and save on unnecessary cost escalations in the future. Using our solutions, our clients deliver more complex, high-quality software, faster and more cost effectively.
At the heart of the SCT solution is the world-renowned Test Management approach (TMap), the company's methodology for structured software testing services. This methodology, built on Sogeti's global experience and expertise, is designed to address the key issues of quality, time and cost. Flexible by design, TMap offers a complete and consistent approach, which is suitable for all types of organizations, of varying size, and can be adapted for use in any development environment. Complementing the TMap approach is Sogeti's Test Process Improvement (TPI) model, a model for reviewing and improving the entire test process.

The SCT solution covers the complete spectrum of software testing and quality assurance. Examples of our TMap related services are:

- Test management
- Testing (specifying and executing tests)
- Test consultancy
- Test automation

- Training and coaching
- Assessment and improvement of the test process
- Setting up a test organization
- Outsourcing of testing (TMap Factories, with local offices in the different countries and back offices in Spain and India)
- Requirements Lifecycle Management
- Reviews and inspections
- Quality assurance.

We offer our services throughout Europe and the USA. Countries where we have local offices are Belgium/Luxembourg, Denmark, France, Germany, Ireland, Spain, Sweden, Switzerland, the Netherlands, United Kingdom, and the USA. The website www.sogeti.com offers connections to all local websites.

In order to be able to continuously follow the developments and trends in IT, Sogeti invests considerably in research & development of the SCT offering. Innovations in technology, new development methodologies, new areas of use and changes in trends of the IT-policies of the leading companies are closely followed.
The results of research & development are published in journals, international newsletters and in TMap Test Topics books, presented on TMap Test Topics seminars and the international test conferences and networks like EuroSTAR, ICSTest and STAREast. Detailed results are also published on www.tmap.net.

Index

TMap Next - Introducing a completely revised and updated TMap

TMap is a test method that can be summarised by four essential points:
- The client manages the test process on rational and economic grounds: Business Driven Test Management (BDTM).
- The comprehensive test process, from test execution through test management, includes ample tips and examples.
- TMap contains a complete 'tool set', i.e. technique descriptions and organisational and infrastructure support.
- TMap is an adaptive method that is suitable for test situations in most environments (such as new development, maintenance, waterfall / iterative / agile development, customised or package software, outsourcing).

TMap offers the tester and the test manager guidelines to deliver results for the client. Through the years, TMap evolved into a de facto standard for testing information systems. It is currently implemented in hundreds of companies and institutions all over the world. The power of TMap can be largely ascribed to the many experiences from actual practice that it incorporates. This makes the book a valuable tool for current and future challenges in the field of testing.

left to right: Rob, Leo, Michiel, Bart and Tim

The authors
Tim Koomen is co-author of the Test Process Improvement-model and won the European Testing Excellence Award 2003. *Leo van der Aalst* is a very experienced international test manager and developer of test services such as outsourcing. *Bart Broekman* is an expert in test design techniques and test methods. He is co-author of Testing Embedded Software and of Test Automation. *Michiel Vroon* is test manager specialised in international organisation affairs and an experienced trainer in testing. *Rob Baarda* has many years of experience as test consultant and was project leader of the book development.

"TMap is one of the most important instruments to control the quality of our IT products."
Nico Jongerius, director ITF, member of general management REAAL Insurances, The Netherlands

"Pick a random spot in this book and you'll find something interesting."
Rex Black, test author and President of the International Software Testing Qualifications Board (ISTQB), USA

"This book will enable a test professional to better balance the test cost with the benefits provided by testing, thus making it easier to get management buy in for the test project."
Ruku Tekchandani, SW Validation Program Manager, Intel Corporation, USA

"TMap is for testing what ITIL is for operational processes."
Wim Waterinckx, Process manager Testing, KBC Group, Belgium

Website: www.tmap.net

ISBN 90-72194-80-2
9 789072 194800