TMap Next

The essentials of TMap

authors: Aalst, L. van der, Broekman, B., Koomen, T., Vroon, M.
based on the original publication in:
TMap NEXT, for result-driven testing,
Aalst, L. van der, Broekman, B., Koomen, T., Vroon, M. (2006),
's-Hertogenbosch: Tutein Nolthenius Publishers, ISBN 90-72194-80-2

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.

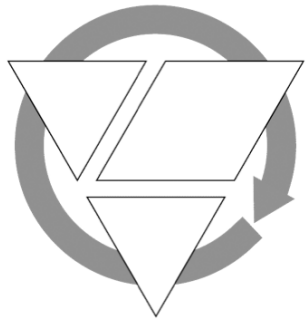

Figure 3.1 TMap model of essentials

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.

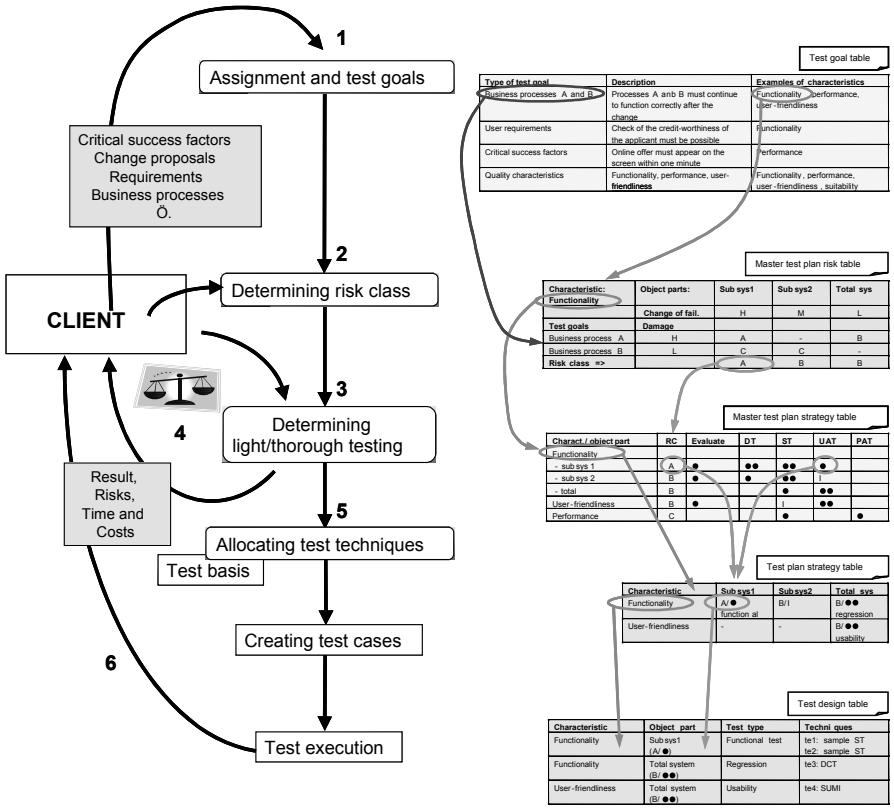

Figure 3.2: BDTM steps

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”).

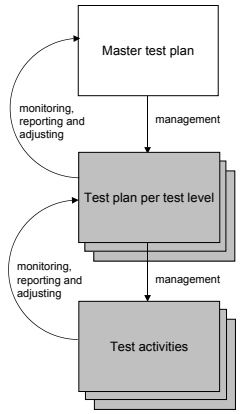

Figure 3.3: Control and test processes

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.

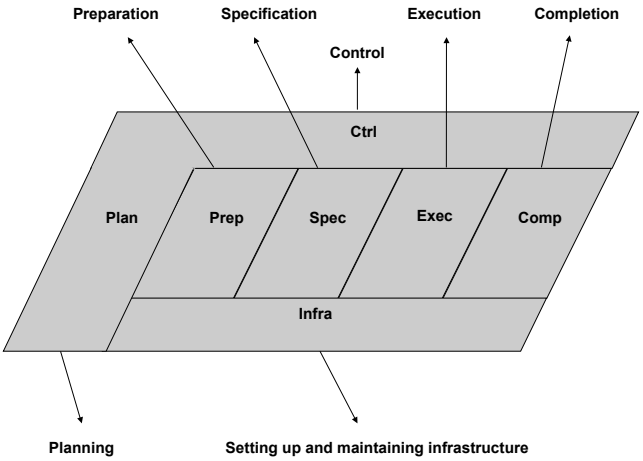

Figure 3.4: TMap life cycle model

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.

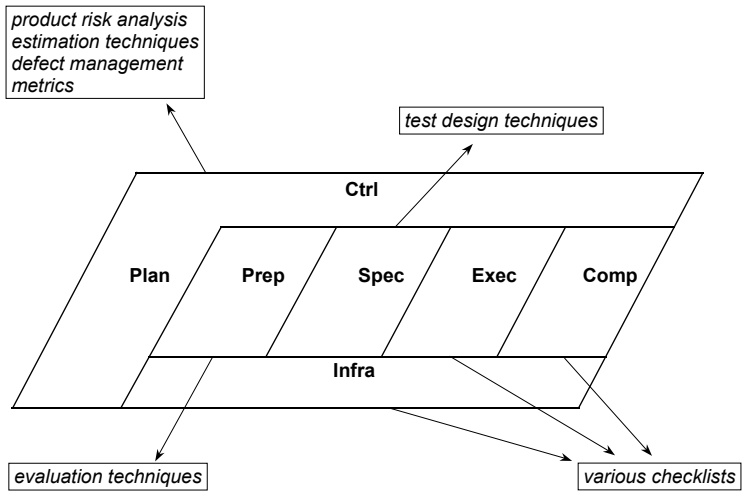

Figure 3.5: Test techniques

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.

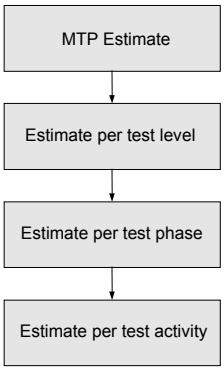

Figure 3.6: Estimation levels

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.

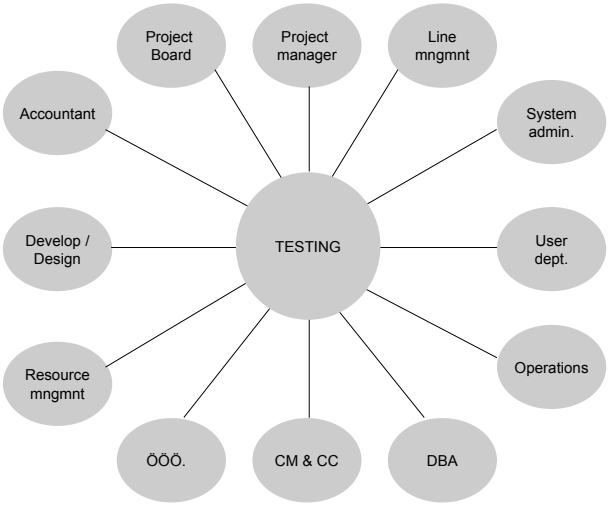

Figure 3.7: The many disciplines involved

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].