EuroSTAR eBook
2017 Series

Test Management

Chapter 3 of

Guide to Advanced Software Testing: Second Edition

Anne Mette Hass

NNIT, Denmark

EuroSTAR Software Testing

biography

contact info:

@AnneMetteHass

Anne Mette Hass has worked in software testing for over 30 years. She is interested in the technical aspect of testing as well as in the human aspect after all, testers are people, people are different, and differences should be understood and respected. Anne Mette has written the book “ uide to Advanced Software Testing” and taught ISTQB Foundation and Advanced Test Manager and Advanced Analyst numerous times. She has spoken at many conferences over most of the world and is fluent in English even though it is not her mother tongue.

abstract

Test management is the art of planning and directing a test project or a test subprocess to success. It is in many ways like project management, and yet not quite the same. The fundamental ISTQB test process is described in Section 2.2. of the Advanced Software Testing book. This process includes the test management activities of test panning, monitoring, control, and test closure. These activities form the basis for the corresponding ISO 29119 test management activities described here, and they are very similar in nature.

This eBook first describes how a test project management interacts “upward” with the development or maintenance project management it belongs to. This is followed by a description of how the test project management interacts “downward” with the management of its individual test subprocesses.

Finally the interaction between the test subprocess management and its dynamic and/or static test processes is described. Following this introduction, the individual activities in the test management process and their outcomes in the form of documents are described.

EuroSTAR Software Testing

abstract

Mappings are provided between documents defined in ISTQB (IEEE 829–1998) and the corresponding documents defined in ISO 29119.

The central detailed activities to be performed in the course of test management—test stakeholder analysis, test estimation, and test progress monitoring and control—are each described in separate sections in order not to confuse the overview of the process provided in Section 3.1.

Likewise, the two central themes of risk-based testing and test metrics and measurement are described in separate sections.

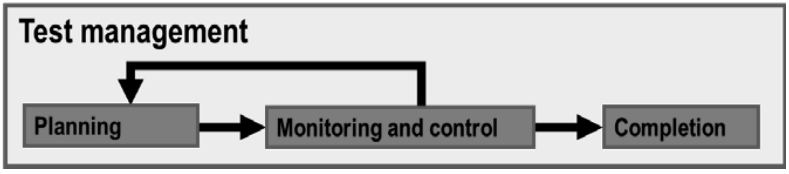
The order of these special sections follows the order in which the activities and themes appear in the course of the test management process. The order is shown in the section overview on the right.

EuroSTAR Software Testing

3.1 Test Management Process

The test management process defined in ISO 29119 can be used for all test projects and test subprocesses, and it includes the following activities:

- Test planning;
- Test monitoring and control;
- Test completion.



The test management process has iterations when monitoring and control give rise to changes in the plan.

The test management process may be applied to a test project as well as to a test subprocess; this is further explained below.

3.1.1 Test Management Responsibility

The responsibility of any test manager or test leader at any level is to plan the work, follow the progress to completion, and change the plan according to reality as the work progresses.

If you fail to plan—you plan to fail!

It is important to plan activities rather than just jump head first into action. The planning work provides a deeper understanding of the task at hand, and it is much easier to change something you have written down or sketched out on a piece of paper than something that has already taken place in the real world.

Planning takes time and because it is so important, it should be planned so that it can start as early as possible. Take your planning seriously, so that you don't end up like this poster painter once did:

The benefits of starting test planning early are many:

- There is time to do a proper job of planning.

- There is more time to talk and/or negotiate with stakeholders.
- Potential problems might be spotted in time to warn all the relevant stakeholders.
- It is possible to be able to influence the overall project plan.

When you plan, you have to keep in mind that a plan needs to be **SMART**:

Specific—Make it clear what the scope is.
Measurable—Make it possible to determine if the plan still holds at any time.
Accepted—Make every stakeholder agree to his or her involvement.
Relevant—Make references to additional information; don't copy it.
Time specific—Provide dates.

All stakeholders for a test plan must agree to the contents according to their interest and involvement—otherwise the plan is not valid!

Remember that *a plan is just a plan*; it is not unchangeable once written. A plan must be a living document that should constantly be updated to reflect what happens in the real world. Contrary to what many people think, it is not a virtue to keep to a plan at any cost—the virtue lies in getting the plan to align with the real world. No matter how hard you try, you are not able to see what is going to happen in the future when you make the plan.

The planned work must therefore be monitored and any deviation from the plan analyzed and acted on, so that the test project or test subproject is still under control. If, for example, an activity takes longer than expected, decisions must be made about what to do: Should we push the deadline? Should we assign more people? Should we reduce the scope? Should we apply a mixture of these measures? Should we do nothing? This is a very important part of the test management responsibility.

3.1.2 Interaction with Other Processes

A test management process interacts with other test management processes at higher and/lower levels and with other processes depending on the context in which it is applied.

Processes cannot communicate with each other as such. The communications will be in written or in oral form between the people filling the various roles in the performance of the processes.

Test management must be done in close cooperation with project management; sometimes by the same person, sometimes by different people, though the two roles should not be filled by the same person if at all possible.

The test manager is the link between the test team and the development team and between the test team and higher management. It is therefore

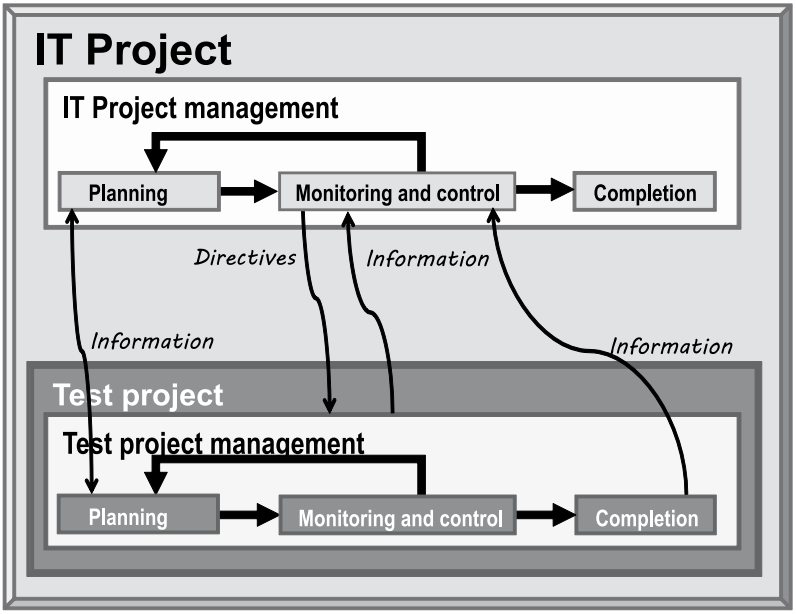
essential for the test manager to be the ambassador of testing and truly understand how testing contributes to the business goals.

However, this does not mean that testers and developers should not talk to each other directly. On the contrary, the more interaction there is between these two groups and between testers and other stakeholders the better. Such interaction facilitates understanding and respect for each other's work and contributes to the common goal of producing an operational and reliable software product.

3.1.2.1 IT Project Management and Test Project Management

Usually a test project is a subproject under a development project or a maintenance project or contract.

The interactions between any form of an IT project management process and a test project management process are shown in the figure below.



The project test plan must be closely connected to the overall project plan, especially in terms of the schedule and budget. The project test plan could either be referenced from or included in the project plan. During initial planning as well as in subsequent updates of the plans, both in the IT project and in the test project, information is shared between the two planning activities to ensure that the two plans are aligned.

Directives will be issued to the test project management during the course of the test project. This could be, for example, in the form of an

announcement from the IT project manager that a specific test subprocess will have to be postponed for whatever reason, or a formal instruction to deliver a test report within a specific time frame.

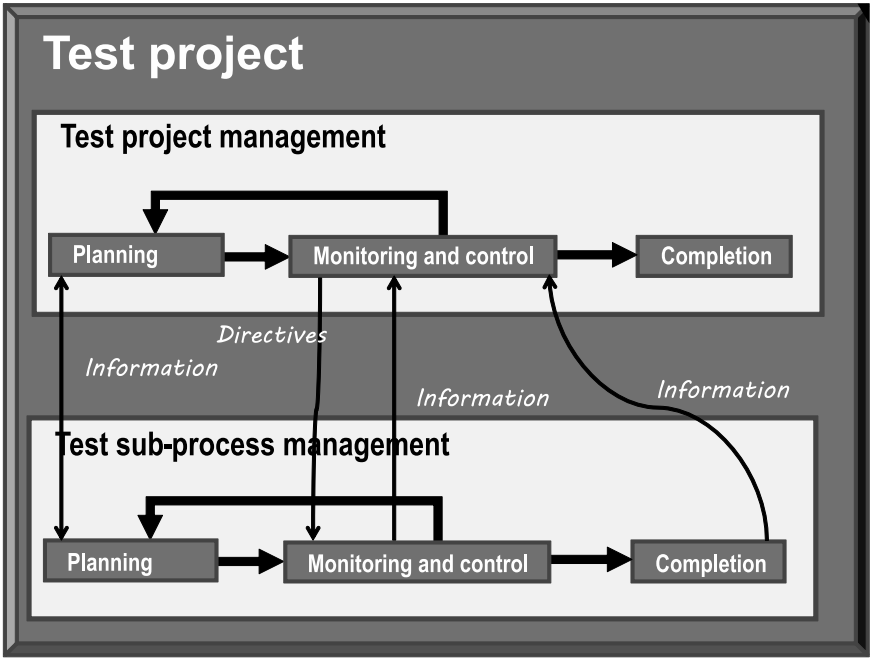
Information will go the other way from the test project to the IT project to keep the IT project manager aware of progress and/or problems with the test project. This information may be used to update the IT project plan, if needed.

When all test subprocesses for a test project are completed, this must be communicated to the IT project.

3.1.2.2 Test Project Management and Test Sub-Process Management

A test project will comprise at least one and usually a number of test sub-processes, depending on the development or maintenance project with which it is associated.

The interactions between a test project management process and any test sub-process management process are shown in the figure below.



During initial planning as well as in subsequent updates of the plans, both in the test project and in any test sub-process, information is shared between the two activities to ensure that the two plans are aligned.

Directives will be issued from the test project management to the test subprocess manager(s) during the course of the test project. This could be in the form, for example, of the test project manager giving the go-ahead for a planned test execution to begin.

Information will go the other way from any test subprocess to the test project to keep the test project manager aware of progress and/or problems with the test subprocess. This information may be used to update the test project plan, if needed. The information may be relayed to the IT project manager as described above.

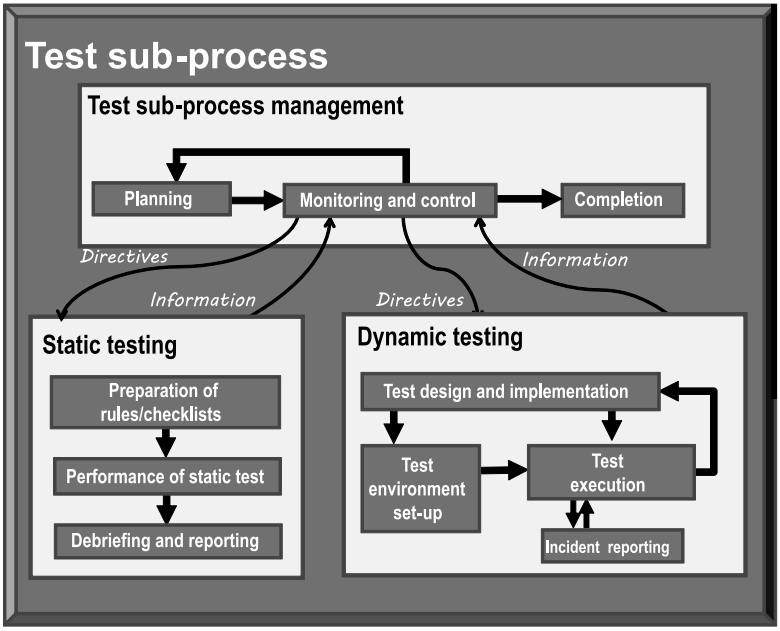
When a test subprocess is completed, this must be communicated to the test project.

Note that there should not be any interactions between individual test subprocess management processes; these should communicate through their test management process, so that the test project manager is informed of what is going on and can make the appropriate decisions.

3.1.2.3 Test Subprocess Management and Static or Dynamic Testing

Static and/or dynamic test(s) will be performed in a test subprocess, depending on the development or maintenance project with which it is associated.

The interactions between a test subprocess management process and a dynamic test process or a static test process are shown in the figure below.



Directives will be issued from the test subprocess management to the dynamic or static test process to initiate or stop action during the course of the test subprocess. This could take the form of, for example, relaying a directive for beginning a planned test execution.

Information will go the other way from any dynamic or static testing process to the test subprocess management to keep the test project manager aware of progress and/or problems with the actual testing. This information may be further relayed and used to update appropriate plans.

When a static or dynamic test is completed, this must be communicated to the test subprocess.

Note that there should not be any interactions between individual static and/or dynamic tests; these should communicate through their test subprocess, so that the appropriate test project manager is informed of what is going on and can make the appropriate decisions.

3.1.3 Detailed Activities

The test management process is in many ways like a classic project management process, only with a specific testing objective. The involved activities are described below.

3.1.3.1 Test Planning

The first activity in the test management process is the test planning. A number of considerations and choices must be made to ensure the most effective and efficient test performance. The results of these choices must be documented as the test plan to guide the test project or test subprocess.

The point is not to create the plan, but to perform the planning activities; that is, to make the necessary decisions about what, how, where, and by whom testing shall be performed.

Test planning requires a firm understanding of the context in which the testing is going to be performed. Relevant material, such as the test policy and/or applicable test strategies, regulatory standards, relevant plans, project documentation, and not least the risk register(s) and the requirements specification(s), must be studied and analyzed.

Creating the testing plan takes time and it should not be "invisible" work (i.e., work that is not scheduled nor reported anywhere). It is part of the planning to include the planning activities in the plan.

The detailed planning activities can be derived from the list of contents of a test plan as suggested, for example, by ISO 29119. This list of contents is described in Section 3.1.4.1.1 below.

Even though the list of contents for a test plan is sequential, the planning activities may be performed in a different sequence and often is an iterative way to make everything fall into place. Consensus of all decisions must be gained from relevant stakeholders.

The test plan, that is, the decisions made as an outcome of performing the detailed planning activities, must be recorded. One purpose of this to be able to remember what the decisions are. Another purpose is to make the decisions available to the relevant stakeholders.

3.1.3.2 Test Monitoring and Control

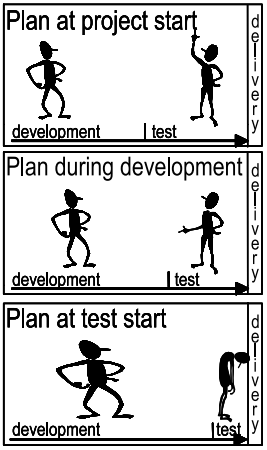
Monitoring and control is an ongoing activity with the purpose of staying in control of the test project or test subprocess at hand. *If you don't control a project, it will control you.*

Monitoring is performed by collecting measurements and comparing these to the plan; a bit like testing actually. If there is any discrepancy, control actions must be planned and implemented and the relevant stakeholders must be informed.

Planning what metrics to collect and how to collect them as well as how to present and use the actual measurements is described in Sections 3.5 and 3.6 below.

Another important part of monitoring is to *revisit the risks*, and identify possible new risks and/or changes in the already identified risks in the risk register(s).

There are a few rules that you must adhere to when following up on the actual activities. Follow up must be guided by:



- Honesty;
- Visibility;
- Action.

First of all you need to be honest, not only when you estimate, but also when you collect information about reality. In the long run you lose integrity and trust if you "tailor" the numbers, or come up with "political" results of the monitoring.

You also need to make the information visible to all stakeholders. Again you lose trust if you hide the truth, whether it is a positive truth (we are ahead of schedule) or a negative truth (we are behind schedule). Information about progress and findings must be made readily available to the stakeholders in appropriate forms, including regular test status reports.

The last thing you need to do to stay in control is to take action whenever needed. *It is your duty as test manager to intervene as soon as deviations occur!*

3.1.3.3 Test Completion

The completion activity is about safeguarding the assets that have been produced during the course of the testing covered by the plan and cleaning up the test environment.

The assets can be both tangible, in the form of, for example, test plans, manual and/or automated test procedures, and/or test environment infrastructure, and intangible, in the form of lessons learned.

Tangible assets should be placed under configuration management such that they are clearly identified, safely stored, and available to those who might benefit from them, for example, for the purpose of regression testing during maintenance.

Lessons learned may be collected at a retrospective meeting. This could be both concerning what went well and what did not go so well during the course of the testing. Recommendations for improvement of both testing and other processes may be derived from the lessons learned and reported to those responsible for process improvement. *In this way the testing not only contributes to better quality in the item under test, but also to better quality in future products and better results for the business.*

3.1.4 Produced Documentation

The tangible outcomes of the performance of a test management process are a test plan and a test completion report. These two types of test management documentation belong to a particular test project typically consisting of a number of test subprocesses. Hence, a complete set of test management documentation might consist of:

- One project test plan;
- One or more subprocess test plan(s);
- A number of test status reports;
- One or more subprocess test completion report(s);
- One project test completion report.

The contents of a test plan, a test status report, and a test completion report as suggested in ISO 29119 are described below.

3.1.4.1 Test Plan

The plan outlines the journey

The project test plan documents the implementation of the relevant organizational test strategy for a particular project. This is where strategy meets reality for the first time. The project test plan must comply as much as possible with the strategy; any noncompliance must be explained, and the reason reported to the person responsible for the organizational test strategy as input for possible changes.

A subprocess test plan documents the details for a specific test subprocess, for example, component testing, security testing, or acceptance testing. The size of a subprocess test plan depends on the test subprocess it describes—a component test subprocess plan for single components may be just 5 to 10 lines; a system test subprocess plan may be several pages.

The structure of the test plans, both the project test plan and any subprocess test plans, should be tailored to the organization's needs.

The way the information in a test plan is presented is not important; the information is. A project may have a project test plan and several subprocess test plans, or all the planning for the entire test project may be held in one document. A test plan does not even need to be a document; the various sections could be kept in appropriate tools, such as spreadsheets or dedicated test management tools.

ISO 29119 suggests the following contents for a test plan. This can, of course, be tailored to the individual situation.

Note that Section 3.1.4.1.2 provides an example of a test plan; you might want to look at this as you read the description of the test plan contents.

3.1.4.1.1 Test Plan Contents

Any document should include some document-specific information. This should adhere to the organization's convention for document identification, if there is one. The information should include the following as appropriate:

- Overview;
- Unique identifier (e.g., title, issue date, and version);
- Issuing organization;
- Approval authority;
- Change history.

Furthermore any document should provide an introduction, which should include:

- Scope of the document;
- References;
- Glossary.

It is important to get the references precise and correct. Testing is influenced by many aspects, including the organization's test policy, the test strategy, the development or maintenance plan, risks, constraints (time, money, resources), and the test basis and its availability and testability. References must be made to all this information, as applicable—and the information must be respected.

If this chapter gets too voluminous you can place some of the contents in appendices.

1. Context of the testing

This section should provide an overview of the test covered in the plan. It may be divided into the following subsections.

1.1 Project(s)/test subprocess(es)

This section should provide a very brief description of the test project or the test subprocess(es) that the plan is covering.

In a test project plan the development or maintenance project to which it belongs should be briefly described. In a subprocess test plan, the overall test project should be mentioned.

1.2 Test item(s)

This section should provide precise and explicit identification of the test item(s). It may also contain additional information such as appropriate basis specification, for example, detailed design or requirements specifications, and helpful information such as design guides, coding rules, checklists, a user manual, and relevant test reports.

It may also describe the business purpose of the test item(s), or reference where this information can be found, as well as any procedures for the transfer of the test item(s) from other environments to the test environment.

The test item can be a software unit, interfaces between units, a subsystem, or a complete system depending on the scope of the test plan.

1.3 Test scope

This section should provide a more specific overview of *the features to be tested*.

A feature to be tested could be a specific attribute or set of attributes of the software, a function, an interface, or a business process. Features include both functional and nonfunctional quality attributes.

The decision about which features are to be tested and which not is based on the applicable test strategy, the identified risks, and the mitigation activities for them. The identification of the features to be tested is also closely linked to the specified coverage items.

To set the expectations of the stakeholders right, it is just as important to state what features are not tested as it is to state which are, and hence a list of the features not to be tested should also be provided. A reason must be given as to why each feature is left out.

1.4 Assumptions and constraints

This section should provide descriptions of any assumptions and constraints for the test covered by this plan. This could be based on the test policy and the applicable organizational test strategy, contract issues, project time and cost constraints, and availability of appropriately skilled staff.

1.5 Stakeholders

Relevant stakeholders should be identified and registered along with useful information about them, so that they can be appropriately involved. Stakeholder analysis is described in Section 3.2 below.

This section should provide the result of a stakeholder analysis for the test project or test subprocess.

2. Testing communication

This section should provide a communication plan for how and when to communicate with different stakeholders during the course of the testing.

The section could include a communication or organizational diagram.

The section may also include overview information about the communication lines in defect handling.

3. Risk register

In risk-based testing some of the most important activities are the risk management activities, including identification of ways of treating the risks. The detailed activities in risk-based testing are described in Section 3.3 below.

This section should provide or reference lists of risks to be considered during the test covered by the plan.

It is a good idea to separate product risks and project risks into two separate risk registers as indicated here.

3.1 Product risks

These are risks related to defects that could cause harm during use if they remain in the test item when it is released. A test item may be released for use in production, for example, the complete system, or released for inclusion in other test items, for example, a component to be integrated with other components.

3.2 Project risks

These are risks related to completion of the test project or test subprocess within the constraints (typically time, quality, and cost).

4. Test strategy

This section should provide an outline of the approach to testing for the specified test project or test subprocess, as outlined in the following subclauses.

This test strategy should be based on the organizational test strategy, which outlines the overall test strategy from a business point of view, and the result of the risk management process as described above (because these are the specific risks and mitigation actions for the scope of the test plan).

This section may be divided into the following subsections.

4.1 Test subprocesses

For a project test plan this section should provide an overview of the test subprocesses that will be performed within the test project.
For a test subprocess plan this section may be omitted.

4.2 Test deliverables

This section should provide an overview of everything that is to be produced and delivered to stakeholders during the course of the planned test. This may typically be in the form of documents or contents in databases, general tools, and/or dedicated test tools.

The overview of test documentation presented in Chapter 2 may provide inspiration for which deliverables a test project or test subproject is expected to produce.

Test input data and test output data may also be identified as deliverables, as well as test tools created as part of the testing activity. If documents have been combined or eliminated, then this list will be modified accordingly.

This subsection may include information about when the document(s) should be delivered, and to/from whom, that is, to which defined role in or outside the project of which the testing is a part.

4.3 Test design techniques

This section should provide a list of the specific test design techniques that are to be applied. References may be made to where the techniques are further described, for example, in a test handbook.

The most common test design techniques are described in Chapter 6 along with general guidelines for selecting which technique(s) to use depending on the risk profile for the test item.

4.4 Test completion criteria

This section should provide a description of what must be in place before the test execution activities for the planned test may be considered to be complete.

This will typically include one or more of the following:

- Specified coverage has been achieved.
- Specified number of failures found per test effort (test hours) has been achieved.

- Number of known outstanding defects is below a specified level (possibly per defect category).
- The benefits of the system are greater than the problems.
- (The time has run out.)

The last one is usually not a very sensible completion criterion; it is nonetheless often encountered in real life, though rarely documented.

4.5 Metrics to be collected

This section should provide a list of the metrics for which measurements are to be collected during the test activities (typically for use in the monitoring activity).

Test metrics are described in Section 3.5 below.

4.6 Test data requirements

This section should provide a list of the test data necessary for execution of the defined test procedures.

This section in the test plan often only provides overall guidelines for the test data related to the test covered by the plan. Detailed test data requirements are identified during the test design activity and may be documented in a separate test data requirements document that may be referred to from here. This document is described in Chapter 6.

4.7 Test environment requirements

This section should provide a list of the necessary and "nice-to-have" properties of the test environment for execution of the defined test procedures. This could include platform(s), middleware, other systems, testing tools, peripherals, and venue.

As for the corresponding section concerning test data requirements, this section in the test plan often only provides overall guidelines for the test environment related to the test covered by the plan. Detailed test environment requirements are identified during the product design and the test design activities and may be documented in a separate test environment requirements document that may be referred to from here. This document is described in Chapter 6.

4.8 Retesting and regression testing

This section should provide a description of the conditions under which retesting and regression testing will be performed.

This could include a description of the estimated number of test cycles, and should refer to the risk profile for the test item.

4.9 Suspension and resumption criteria

This section should provide a description of any criteria used to suspend and resume all or some of the testing activities. It should also identify who has the authority to suspend and who can resume testing activities.

If test activities have to be repeated when testing is resumed, this should also be specified.

This section may be included in Section 3.2, project risks of the test plan.

4.10 Deviations from the organizational test strategy

This section should provide a list of any deviations from the organizational test strategy introduced in this test plan, along with the approving authority/authorities.

5. Testing activities and estimates

This section should provide an overview of *ALL* activities to be performed for the test covered by the plan. This may be presented in the form of a work breakdown structure, possibly using a general planning or a dedicated test planning tool.

Time and possibly cost estimates should be provided for the activities. Estimation is further described in Section 3.4 below.

Sometimes budget and cost issues are only dealt with in the project plan or in the test project plan.

The detailed overview of activities and estimates for these will evolve as the test progresses. It is not possible, and should not be attempted, to provide the same level of detail for activities just about to be performed and activities to be performed in months or even years into the future.

6. Staffing

This section should provide an overview of the roles that are needed in the test project or test subprocess, and the organizational unit and/or people who are filling these roles.

The preparation of this section is often iterative, going through the detailed descriptions of the role, identifying staff, possibly hiring and training new people, and assigning people to roles several times before the puzzle is completed.

6.1 Roles, activities, and responsibilities

This section should provide a list of the roles needed for the test at hand. This will, of course, include testing roles, but it may also include other roles, for example, operations staff, user representatives, technical support staff, data administration staff, and quality support staff.

A role description including required and possibly also "nice-to-have" skills should be provided or referenced for each role.

The number of people needed for each role should also be identified, for example two test analysts and one half-time technical test analyst.

The list of roles is used to staff the test project; that is, to assign organizational unit and/or named people to roles and define their specific responsibilities. It should also be used to identify staff recruitment and/or training needs.

The result of the identification of activities and staffing can be described in a RACI matrix (responsibility assignment matrix). RACI is short for the responsibilities that can be assigned to roles, namely:

Responsible—the role(s) performing the activity;
Accountable—the role carrying the economic liability;
Consulted—the role(s) that may contribute or influence;
Informed—the roles that must know about the activity and its results.

In a RACI matrix, the activities are often listed vertically, the roles horizontally, and the RACI marked in the intersecting cells.

6.2 Hiring needs

This section should provide a list of the people who may have to be hired, transferred, contracted, or otherwise acquired to fill roles that may not be otherwise filled. It must describe when people are needed and for how long, if not permanent full time. Necessary and wanted skills should also be expressed as precisely as possible.

6.3 Training needs

This section should provide an overview of any training required for the people filling the roles in the test project. The training should be part of the activities to be handled in the schedule.

7. Schedule

This section should provide the detailed schedule created from the tasks and their dependencies, the staffing, and the estimates. This should provide the expected start time and duration for each testing activity and associated test milestones.

This is where it is shown when and for what amount of time each person is allocated to perform which activities.

This is the most versatile part of the plan, since the schedule will change over time, when tasks are being performed and estimates are refined.

3.1.4.1.2 Project Test Plan Contents Example

The table below provides some examples of contents for each of the entries in a specific instantiation of a test plan. The examples are rather short, but I hope they may serve as inspiration.

Note that text in brackets refers to examples in this book.


System Test Plan	
Context of the testing	
Project	This system test is part of the KLOD project.
Test item(s)	The completed KLOD product delivered for system test. This includes the software and the documentation.
Test scope	All functional and nonfunctional requirements specified in the KLOD requirement specification must be covered in this test, except the performance requirements because the dynamic performance test will be carried out by third-party company PTESTIT.
Assumptions and constraints	This plan is based on the constraint that the deadline for decision on go/no go based on the system test is a kill or cure deadline and will be maintained, no matter how far the system test has proceeded by then.
Stakeholders	The identified stakeholders are: (See result of stakeholder analysis in Section 3.2.2.)
Testing communication	
(See result of stakeholder communication analysis in Section 3.2.3.)	
Risk register	
Product risks	(See examples of product risks in Section 3.3.1.2.3.)
Project risks	(See example of project risks in Section 3.3.1.2.4.)
Test strategy	
Test deliverables	–This test plan; –Test specification; –Test data requirements document; –Test environment requirements document; –Specified test environment, including test data; –Logs and test status reports; –System test completion report.
Test design techniques	The following test design techniques must be used, depending on the nature of the each requirement: –Classification tree; –Decision table; –State transition.

Test completion criteria	–100% requirements coverage; –80% test element coverage, as per applied test case design technique for the requirements; –0 known failures of category A; –Max. 2 known defects of category B; –Max. 15 known defects of category C.
Metrics to be collected	(See Section 3.5.)
Test data requirements	The required test data will be documented in a specific test data requirements document that will be produced during the test analysis and design process.
Test environment requirements	The required test environment will be documented in a specific test environment requirements document that will be produced during the test analysis and design process.
Retesting and regression testing	All test cases that have exposed failures must be rerun when the underlying defect(s) has/have been reported, corrected, and component tested. An analysis must be performed and documented for each defect correction to determine the appropriate regression testing to be executed.
Suspension and resumption criteria	Test execution may be suspended if more than 20% of the execution time is spent reporting banal failures, caused by defects that should have been found during component testing. Development must report back when they have performed sufficient component testing. At resumption, 30% of executed test procedures and the same number of as-yet unexecuted test procedures must be executed to ensure that the quality of the test item has improved.
Testing activities and estimates (A work breakdown structure is too voluminous to include here, but it can be derived from the process descriptions.)	
Staffing	
Roles, activities, and responsibilities	This is illustrated in the RACI matrix shown here:

	1	2	3	4	5	6	7
Test manager	A	A	A	I	A	A	A
Development manager	C	C		A	I		I
Test analysts	C	R	I	I	R	R	R
Technical test analysts	C	R	R	I	R	R	R
Quality assurance	I						I
Sales/marketing	I						I
The customer	I						I
Method department	I	C/I	C/I				I

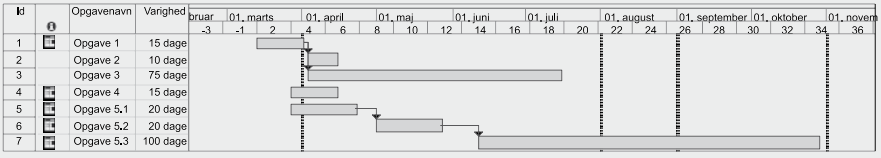
1. Test management
2. Test design
3. Test tools
4. Test environment
5. Test data
6. Test execution
7. Reporting

Responsible **A**ccountable **C**onsulted **I**nformed

Hiring needs	We have to hire a technical analyst to start by September 1 at the latest to fill the technical analyst role.
Training needs	All the staff assigned to this system test must have participated in the half-day introduction to KLOD system test.

Schedule

The schedule is maintained in the ProfProject tool under KLOD, System Test. This is just an example of a Gantt diagram.



3.1.4.1.3 Test Plan Contents Mapping ISTQB/ISO 29119

The ISTQB syllabi mention a master test plan and refer to IEEE 289–1998 for contents. The table below therefore presents a mapping between the widely used IEEE description of the contents of a test plan and the table of contents provided in ISO 29119. A blank entry means that there is no equivalence.

Note that IEEE 829 is superseded by ISO 29119.

ISTQB (IEEE 829–1998)	ISO 29119:2013
Test plan identifier	Unique identification of document
Introduction (scope, risks, and objectives)	Introduction (Scope, References, Glossary)
	Context of the testing –Project/test subprocess –Test item(s) –Test scope –Assumptions and constraints –Stakeholders
Test item(s) or test object(s)	(In context, test items)
Features to be tested	
Features not to be tested	
Approach (must at least cover): –The test methods and test techniques to use –The structure of the test specification to be produced and used –The tools to be used –The interface with configuration management –Measurements to collect –Important constraints	Test strategy: –Test subprocesses –Test deliverables –Test design techniques (Test environment requirements) (Metrics to be collected) –Test completion criteria –Metrics to be collected –Test data requirements –Test environment requirements –Retesting and regression testing –Suspension and resumption criteria –Deviations from the organizational test strategy
Item pass/fail criteria (exit criteria including coverage criteria)	(Test strategy: Completion criteria)
Suspension criteria and resumption requirements	(Test strategy: Suspension and resumption criteria)
Test deliverables (work products)	(Test strategy: Test deliverables)
Testing tasks	Testing activities and estimates
Environmental needs	(Test strategy: Test data requirements, and Test environment requirements)
Responsibilities	(Staffing: Roles, activities, and responsibilities)
Staffing and training needs	Staffing –Roles, activities, and responsibilities –Hiring needs –Training needs
Schedule	Schedule
Risks and contingencies	Risk register –Product risks –Project risks
Test plan approvals	Approval authority

3.1.4.2 Test Status Report

The purpose of the test status report is to provide information to relevant stakeholders about the status of the testing that is performed in a specific reporting period.

Status reports should be issued according to the communication plan outlined in the testing communication section in the relevant test plan.

In a test project or test subprocess, a number of different test status reports may be issued depending on the target audience. The people working directly on the test activities may need daily or very frequent, very detailed status reports, whereas management may need less frequent and less detailed status reports.

ISO 29119 suggests the following table of contents for a test status report. This may, of course, be tailored to the individual situation.

Note that Section 3.1.4.2.2 provides an example of a test plan; you might want to look at this as you read the description of the test status report contents.

3.1.4.2.1 Test Status Report Contents

Any document should include some document specific information. This is described in Section 3.1.4.1.1.

1. Test status

   1.1 Reporting period

   This section should provide information about the period over which the measures presented in the report have been collected, that is, the time period covered by the report.

   1.2 Progress against test plan

   This section should provide an overview of how the testing has progressed in the covered period compared to the schedule.

   If notable deviations are found, these should be listed with the following information:

   - Explanations of the reasons for deviation;
   - Description of any actions;
   - Description of the effects;
   - Implications with regard to planned project objectives.

   1.3 Factors blocking progress

   This section should provide an overview of the factors that have impeded progress during the reporting period, if any, with descriptions of how the obstacles have been overcome. Any unsolved issues should also be listed with suggested solutions.

1.4 Test measures
This section should provide suitable presentations of the test measures collected during the reporting period and relevant at the end of the reporting period.

This information may be presented graphically. For more details see Section 3.6 below.

1.5 New and changed risks
This section should provide the result of a renewed risk identification and analysis; that is, new risks and updated remaining risks.

1.6 Planned testing
This section should provide a description of the test activities planned for the next reporting period. This information is usually presented as an updated schedule.

3.1.4.2.2 Test Status Report Contents Example

The table below provides some examples of contents for each of the entries in a test status report. The examples are rather short, but I hope they may serve as inspiration.

System Test Status Report	
Test status	
Reporting period	Week 4, 2013 (21.1–25.1)
Progress against plan	All the test procedures were executed according to the plan; except TP 31.C and TP 31.D. These have been postponed to next week, when there should be enough time to execute them.
Factors blocking progress	The test environment concerning the interface to HUNF was delayed 5 hours on Wednesday. We have found an unexpected high number of trivial defects in the report creation functionality. This has caused some delay.
Test measures	(See examples below in Section 3.6.)
New and changed risks	No new risks have been identified. Risk R45 has been removed based on the results of executing PR16.
Planned testing	See updated test schedule on the project wall.

3.1.4.2.3 Test Status Report Contents Mapping ISTQB/ISO 29119

The ISTQB syllabi mention a test report and refer to IEEE 289–1998 for contents. IEEE 829–1998 does not include a test status report and hence there is no mapping possible for the ISO 29119 test status report contents.

3.1.4.3 Test Completion Report

The purpose of test reporting is to summarize the results of a test project or test subprocess and provide evaluations based on these results.

A test report should be issued at the completion of each test level and the end of the entire testing assignment task. The test reports should include analysis of result information to allow management decisions, based on risk, about whether to proceed to the next level of testing or to project implementation, or whether more testing is required.

Top management or other stakeholders may also need test completion reports at other points in time, for example, for regularly scheduled project status meetings or at the end of the project in order to adjust policy and strategy.

ISO 29119 suggests the following table of contents for a test completion report. This may, of course, be tailored to the individual situation.

Note that Section 3.1.4.3.2 provides an example of a test plan; you might want to look at this as you read the description of the test status report contents.

3.1.4.3.1 Test Completion Report Contents

Any document should include some document specific information. This is described in Section 3.1.4.1.1.

1. Testing performed

    1.1 Summary of testing performed

    This section should provide an overview of the testing performed for the test project or test subprocess covered by the report. The section could refer directly to the test plan. It should be possible to read the summary in isolation and get the main information about the test.

    1.2 Deviations from planned testing

    This section should provide an overview of the deviations from the planned testing, if any, including what was done and not done compared to the original plan.

    1.3 Test completion evaluation

    This section should provide an account of whether or not the original (or modified) plan was followed to completion. It should describe which of the planned tests were not performed, if any, and why not.

This is also where a description goes of how the original completion criteria were met. If the completion criteria were modified during the course of the testing, an explanation should be provided here.

Any statistically valid conclusions that can be drawn from these analyses could be used to predict the quality level achieved in the tested product.

1.4 Factors that blocked progress
This section should provide an overview of the factors that have impeded progress during the reporting period, if any, with descriptions of how the obstacles have been overcome.

1.5 Test measures
This section should provide suitable presentations of the test measures collected during the course of the testing.
This information may be presented graphically. For more details see Section 3.6 below.

1.6 Residual risks
This section should provide a list of the risks that have not been resolved at the end of the testing; this may be risks that have not been fully treated by the test and/or any new risks identified as a result of the final monitoring and closure of the test.

1.7 Test deliverables
This section should provide a list of all the test deliverables produced as a result of the test effort and where they are stored. The list should be compared to the corresponding list in the related test plan, and any discrepancies explained.

1.8 Reusable test assets
This section should provide a list of the test deliverables and other artifacts produced during the course of the testing that can be reused. The artifacts should be placed under configuration management or handed to the appropriate stakeholders for storage.

1.9 Lessons learned
This section should provide the result of any lessons learned meetings in a form that is suitable for the organizational unit in charge of process improvement and/or any other relevant stakeholder(s).

3.1.4.3.2 Test Completion Report Contents Example

The table below provides some examples of contents for each of the entries in a test completion report. The examples are rather short, but I hope they may serve as inspiration.

System Test Completion Report	
Test performed	
Summary of testing performed	Activities planned in the system test plan have been performed. We have executed 231 of the 254 specified test procedures. 108 incidents have been reported, hereof –2 category A defects, –11 category B defects, –38 category C defects. The rest are category D defects and misunderstandings. 2 defects were identified in the test procedures; these have been corrected and the test procedures reexecuted.
Deviations from planned test	Due to the delay in setting up the test environment for the interface to the HUNF, the execution of the related test procedures was postponed. This was remedied by executing other test procedures earlier than planned.
Test completion evaluation	We have reached the coverage completion criteria, and the numbers of remaining defects are below the maximum allowed number per category.
Factors blocking progress	None that were not overcome during the test.
Test measures	(See examples in Section 3.6 below.)
Residual risks	Three risks remain; see the Product Test Register in the project room.
Test deliverables	All documentation specified in the system test plan has been delivered. See more details in the project room.
Reusable test assets	The database containing the test data used in this system test has been stored under configuration management for reuse in acceptance testing and possible testing during production and maintenance of the system.
Lessons learned	A report has been delivered. The main lesson learned is that more testing should be performed in the component test step, because about 25% of the defects found in this system test could have been found more efficiently in a more thorough component test.

3.1.4.3.3 Test Plan Contents Mapping ISTQB/ISO 29119

The ISTQB syllabi mention a test summary report and refer to IEEE 829–1998 for contents. The table below therefore presents a mapping between the IEEE description of the contents of a test summary report and the table of contents for a test completion report provided in ISO 29119. A blank entry means that there is no equivalence.

Note that IEEE 829 is superseded by ISO 29119.

ISTQB (IEEE 829 - 1998)	ISO 29119:2013
Test plan identifier	Unique identification of document
Summary of testing performed	Summary of testing performed
Variances	Deviations from planned testing
Comprehensiveness, assessment	Test completion evaluation
Summary of results	
Evaluation	
Summary of activities	
	Factors that blocked progress
	Test measures
	Residual risks
	Test deliverables
	Reusable test assets
	Lessons learned

3.2 Test Stakeholder Analysis

One of the rules in warfare is "Know your enemy." Product development and testing is not warfare, and no enemies are involved, but the rule can be modified to "Know your stakeholders," which is just as crucial for success.

A test stakeholder is anybody who will in any way be affected by or will affect the testing of the new product under development. A stakeholder can be a person, an organizational unit, a group of people with similar stakes, or another product.

Stakeholders may have a positive influence; that is, they may be beneficial to the testing process, for example, by providing money and/or expertise. Stakeholders may also have a negative influence; that is, they could represent a risk to testing, for example, by withholding resources.

It is therefore important to know who the stakeholders for a test project or test subprocess are, and how they can be communicated with and drawn into the work as appropriate.

3.2.1 Identifying Test Stakeholders

Basically the way to identify test stakeholders is to ask in relation to the test and the product to be produced and used:

- WHO knows;
- WHO feels;
- WHO benefits;
- WHO suffers.

Test stakeholders can come from any part of the business and the future user realm.

Techniques for identifying the stakeholders for a project may include:

- Checklists from previous projects;
- Interviews;
- Brainstorm meetings;
- Document studies.

Think far and wide when identifying test stakeholders. It is a good idea to "go overboard" and identify too many potential stakeholders, and then discard the irrelevant ones, rather than run the risk of missing somebody.

The following table may be used for inspiration:

Area	**Possible Stakeholders**
The business	Top management Product line manager Product sponsor Business analysts Finance executives Lawyers Marketing Sales Customers
Users	Users of existing product(s) to be replaced Future users More or less official user groups Future users' management Indirect users (users of the results from the product)
Development	Project management Requirements engineers Architects Designers Programmers
Support processes	Process managers Configuration managers Quality assurance User trainers
Production and maintenance	Operations management Operation personnel Support personnel

List all possible stakeholders by name of people or organizational unit or by role, and expand the list with relevant information as described below.

3.2.2 Test Stakeholder Register

The identified stakeholders should be listed in a stakeholder register, and relevant information provided for each. The relevant information, apart from the name of the stakeholder, could include:

- Contact information;
- Role;
- Responsibilities beyond those implied by the role;
- Authority (level of power);
- Rights;
- Availability;
- Relevance;
- Knowledge areas and depth of knowledge;
- Personal goals and interests;
- Possible positive influences;
- Possible negative influences;
- Possible contributions;
- Communication needs.

The list provided here is fairly extensive; the relevant pieces of information for the test stakeholders should be determined for any given test situation.

The list should be reviewed regularly and kept up to date. Information about test stakeholders may change quickly, and an outdated stakeholder register is a potential risk to the test project.

3.2.3 Communication Plan

Stakeholders should be involved as much as possible, both to benefit from their interest and knowledge and to mitigate any opposition toward the product and/or the testing.

It is a very good idea to develop a plan for communication with stakeholders to ensure that each of them gets the information they need in the right format at the right time. Communication with stakeholders can be the "all or nothing" for the success of a project.

The table below shows some examples of communication planned for various test stakeholders.

Stakeholder	Communication
Project manager	Weekly progress meetings Monthly test status report Test completion reports
User representative	Tailored test completion reports
Programmers	Daily overview of new and corrected defects during test execution periods
Testers	Daily overview of new and corrected defects during test execution periods
Quality assurance	Test completion reports

3.3 Risk-Based Testing

The golden rule of testing is:

Always test so that whenever you have to stop you have done the best possible test.

Everybody with some understanding of requirements and testing will know that a requirement like this cannot be verified. What does it mean: the best possible test? It is not immediately measurable.

What is the best possible test then? The answer to that is: It depends!

The best possible test depends on the risk associated with having defects left in the product when it is released to the customer!

The best possible test is determined by the risks we are facing and the risks we are willing to run. Obtaining consensus from stakeholders on the most important risks to cover is essential.

3.3.1 Introduction to Risk-Based Testing

We have to live with the fact that it is impossible to test everything. Testing is sample control. There is a risk involved in all sample control: the risk of overlooking defects in the areas we are not testing.

3.3.1.1 Risk Definition

A risk is defined as "The possibility of realizing an unwanted negative consequence of an incident."

Alternatively a risk may be defined as *"A problem that has not materialized yet and possibly never will."*

There are two important points in these definitions:

- A risk entails something negative.
- A risk may or may not happen—we don't know.

A risk therefore has two aspects:

- Effect (impact—consequence);
- Probability (likelihood—frequency).

The two aspects of risk can be combined into:

Risk level = probability × effect

From this it is quite clear that if we have no probability or no effect we have no risk. The risks that do have a risk level greater than zero are the risks we have to deal with.

It is not a risk (to us), if there is no effect of an event that might happen. We can therefore ignore it even if the probability is high.

There is a probability that there are defects in the new version of our database management system, but that will have no effect on the quality of our product if that does happen, because it is not used in our system.

It is not a risk if there is no (or an extremely small) probability that an event will happen, even if the effect would be extremely big, if it did. We can therefore ignore that as well.

There is no (detectable) risk of our department closing down, because we have lots of orders, are making good money, and both management and employees like their jobs. If we did close down, the effect on the project would be pretty bad, if not disastrous.

It is not a risk either if the probability of an event with a negative effect is 100%. In this case we know we have a real problem on our hands, and we will have to deal with that in our planning.

It is a problem—not a risk—that we will have to do without one of our test experts because she has found another job and is leaving in 3 months.

3.3.1.2 Risk Types

It is quite common to treat all the risks we can think of in connection with a development project in one big bundle. This can be quite overwhelming.

It is therefore a very good idea to take a closer look at the risks and divide them into classes, corresponding to where they may hit—or what they are threatening.

Risks hit in different places, namely:

- The business;
- The processes;
- The project;
- The product.

3.3.1.2.1 Business Risks

The business risks threaten the entire company or organization from a "staying-in-business" point of view. This discussion is beyond the scope of this book and is not discussed further.

3.3.1.2.2 Process Risks

Process risks are related to the processes and/or the way work is performed. This is also beyond the scope of this book, but will be briefly discussed because knowledge about processes is indispensable in a modern development organization, and because testing is also performed according to processes.

Process risk threatens the effectiveness and efficiency with which we work on an assignment. Process risks may originate from:

- Missing process(es);
- The organization's lack of knowledge about the processes;
- Inadequate processes;
- Inconsistencies among processes;
- Unsuitable processes;
- Lack of support in the form of templates and techniques.

Process risks jeopardize the way the work in the project is being performed; including the way testing is being performed. These risks should be the concern of the project manager and/or the test manager, and those responsible for the processes in the organization.

Process risks may influence business risks, as well as project and product risks.

3.3.1.2.3 Project Risks

Project risks are related to the project and the successful completion of the project.

A project consists of a number of activities and phases, from requirements development to the final acceptance test. These activities are supported by activities like quality assurance, configuration management, and project management.

Activities in a project are estimated, get resources allocated, and are scheduled. As the project progresses the activities are monitored according to the plan.

Risks concerning the project may originate from:

- People assigned to the project (e.g., their availability, adequate skills and knowledge, and personalities);
- Time (deadlines);
- Money;
- Development and test environment, including tools;
- External interfaces;
- Customer/supplier relationships.

Project risks jeopardize the project's progress and successful completion according to the plan.

Examples of project risks include the following:

- The necessary analysts may not be available when the requirements development is expected to start.
- Two of the senior designers are not on speaking terms and useful information exchange between them is not happening—this may cause the design phase to take longer than expected.
- The complexity of the user interface may have been underestimated.
- The testers may not be adequately trained in testing techniques, so testing may require more resources than expected.
- The integration may be more time consuming than expected.
- The access to external data may not be possible with the technique chosen in the design.

The project risks are the main concern of the project manager and higher management.

The test manager is concerned with the project risks related to the test project as it is specified in the test plan.

3.3.1.2.4 Product Risks

Product risks are related to the final product. They are the risks of defects remaining in the product when it is delivered.

The risks threatening the product are the testers' main concern. This is where we can make a difference.

We want to deliver the required quality and reliability. This cannot be tested into the product at the end of the development, but must be worked into the product through the work products produced during development and in the implementation of the components. Product risks may be originated in functional and nonfunctional requirements in the form of, for example:

- Missing requirements;
- Ambiguous requirements;
- Misunderstood requirements;
- Requirements on which stakeholders do not agree.

Research (for example, from IBM) has shown that more than 50% of all defects in products can be traced back to defects in the requirements.

Product risks jeopardize customer satisfaction, and maybe even the customer's life and livelihood.

Product risks may be related to different requirement types, such as functionality, safety, security, and political and technical factors.

Examples of product risks are:

- A small, but important functionality may have been overlooked in the requirements and is therefore not implemented.
- A calculation of discounts may be wrongly implemented and the customer may lose a lot of money.
- The instrument may reset to default values if it is dropped on the floor.
- It may be possible to print a report of confidential customer information through a loophole in the reporting facility.
- The installation procedure may be difficult to follow, and this may lead to incorrect and/or incomplete installation.

Project risks and product risks can influence and be the cause of each other. A project risk may cause a product risk, and a product risk may cause a project risk. Risk mitigation actions for a risk may also introduce new risks and/or increase (or indeed decrease) the risk levels of other risks.

If a project risk results in time being cut from component testing, this may cause the product risk of an increased number of defects remaining in the components that are not tested or not tested sufficiently. This may further cause the project risk that there may not be sufficient time to perform a proper system test because too many trivial failures are encountered in the system test.

3.3.1.3 Testing and Risk Management

Test and management of risks should be tightly interwoven as they support each other. In risk-based testing, the test strategy and plan are based on the results of risk analysis, and test results give valuable feedback to support continuous risk analysis.

The result of a product risk analysis can be used in the test planning to make the test as effective as possible. It can be used to target the testing effort, since different types of tests are most effective for different risks.

Component testing is most effective for testing in the case where the product risk level related to complex calculations is highest.

The risk analysis results can also be used to prioritize and distribute the test effort. The areas with the highest risk level should be planned to be tested first and given the most time and resources.

Finally the product risk analysis can be used to qualify the testing already done. If the test effort is related to the identified risks, it should be possible to report on resolved and remaining risks at any time.

Testing can, as mentioned above, resolve or mitigate product risks. The probability of sending a product with defects out to the customers is reduced by the testing finding failures and the subsequent correction of the defects.

Testing can also mitigate project risks if an appropriate test strategy is applied, especially if testing is started early.

Even process risks may be reduced by analyzing failure reports and taking appropriate process improvement initiatives.

3.3.2 Risk Management

Risk management consists of the following activities:

- Risk identification;
- Risk analysis;
- Risk mitigation;
- Risk follow-up.

In risk identification we are finding out what may be threatening the process, the project, or—in this particular context—customer satisfaction with the product.

The identified risks are evaluated in the risk analysis and ordered relative to each other according to their level. The analysis means that we assign probability and effect to the risks, and based on this we can determine the risk level and hence which risk is the worst and how the others relate to that.

One of the points in risk management is to use the results of analyses to mitigate the risks. Actions can be planned to lower the probability and/

or the effect of the risks. In the context of product risks and testing, test activities can be planned to mitigate the risks by lowering the probability of having remaining defects in the product when it is released. The more defects we can remove from the product as a result of the testing, the more the probability falls. *Testing cannot change the effect of a risk.*

Contingency planning is a part of classic risk management, but this is not relevant for product risks in relation to testing. Testing is concerned with lowering the probability of defects remaining in the product. For the defects that still remain when the product is in production, support and maintenance must be prepared to provide work-arounds and/or corrections and updates.

Risk identification, analysis, and mitigation should not be a one-time activity. It is necessary to follow up on the risks as testing progresses. The results of the testing activities provide input to continuous risk management.

Information about the failure frequency over time can be used to assess if the probability of a risk is falling or rising.

3.3.2.1 Risk Identification

Risk identification is finding out where and how things might go wrong, and writing it down to form the basis for risk analysis.

Risks are found in areas where the fulfillment of expectations may be threatened, that is, where the customer satisfaction is jeopardized. *Satisfying expectations is the way to success!*

All stakeholders have expectations about the product, but we are most concerned with the expectations of the customer. The customer orders the product, pays for it, and takes advantage of it—the latter possibly through end users.

In an ideal world the customer's expectations are expressed requirements. These requirements are transformed into a design, and the requirements are fulfilled in the code and or in other types of subsystems being integrated into the final products.

In lesser ideal worlds expectations may be derived from other sources.

Risks are not always evident. Even when we work with experienced and knowledgeable stakeholders, it can be efficient to use a risk identification technique.

Useful techniques include:

- Lessons learned;
- Checklists;
- Risk workshops;
- Brainstorming;

- Expert interviews;
- Independent risk assessments.

Techniques may be mixed to be even more efficient.

3.3.2.1.1 Lessons Learned and Checklists

Lessons learned and checklists are closely related. A checklist is a list of generic risks, formed and maintained by experience (i.e., lessons learned from previous projects).

Risk checklists are valuable assets in an organization, and should always be treated as such.

One or more product risk checklists should be kept in the organization, depending on the diversity of the nature of the projects performed by the organization.

The checklists used by pilots before take-off are long and must be run through very carefully before every take-off. A pilot was once hurried along by a busy business man who asked him to drop the checklists and get going. The pilot carried on with his work as he answered, "These checklists are written in blood!"

3.3.2.1.2 Risk Workshops

Workshops are an effective way to identify risks. There are no strict rules as to how a workshop should be conducted, but a few guidelines can be given.

As many stakeholders as possible should be involved, though the number of participants should not exceed 10 to 12 in order to give everybody a chance to talk within reasonable intervals.

Risk workshops can get emotional and a neutral facilitator—somebody who is not by any account a stakeholder in the project—should be present to guide the discussions. *Encourage discussions and new ideas, but avoid conflicts.*

Make sure that all participants agree that the objective of the workshop has been reached and that it is clear how work can proceed after the workshop.

3.3.2.1.3 Brainstorming

A brainstorming session (in this context of risk identification) is an informal session designed to identify possible risks connected with the product when it is released.

The only rule that should apply during brainstorming is *that no possible idea must be criticized* in any way by the participants. Ideas should be allowed to flow freely, the rationale being that even the most seemingly stupid, silly, or strange thought may be the inspiration for valuable potential risks.

A brainstorming sessions must have a facilitator who can act as a catalyst if ideas do not flow freely. At the end of the session, the facilitator must make sure that whatever surfaced as possible risks is documented.

Any type of stakeholder may be involved in brainstorming.

3.3.2.1.4 Expert Interviews

Interviews may be conducted as individual interviews or as group interviews. *An interview is not as easy to conduct as many people think.* It requires specific skills and thorough preparation to get as much information as possible from an interview.

First of all, an interview is not like an ordinary conversation. People in an interview have different roles (e.g., the interviewer and the interviewees), and they may have a number of expectations and prejudices related to these roles. Interviews must be prepared. The interviewer must make sure, for example, that the right people are being interviewed and the right information is being gathered. A list of questions or a framework for the course of the interview must be prepared.

Ample notes must be taken and/or the interview can be recorded (with the permission of the interviewees). *The interviewer should extract a list of possible risks from the interview* and get agreement from all the participants.

3.3.2.1.5 Independent Risk Assessments

In cases where conflicts are threatened, external consultants may be called in to identify risks. External consultants could also be used if time is short or if specific expertise is not present within the immediate stakeholders.

The external consultants identify risks and usually also perform or facilitate the risk analysis.

The consultants may be external to the project organization or third-party consultants entirely external to the developing organization.

3.3.2.2 Risk Analysis

Risk analysis is the evaluation of the identified risks. One thing is to identify and list the risks; another is to put them into perspective relative to each other. This is what the analysis of the risks helps us do.

The analysis must be performed by all appropriate stakeholders, because they all have different perspectives, and risk analysis is aimed at providing a common and agreed-on view of the risks. Experts may be called in to contribute if adequate expertise cannot be found among the immediate stakeholders.

Risk analysis can be performed more or less rigorously, but it should always be taken seriously.

3.3.2.2.1 Risk Register

A risk register is a very useful tool in risk management. It can be used to support risk analysis and risk mitigation.

Risk registers can be created using office tools, for example, spreadsheets that support calculations.

A risk register should include:

- Risk identification (e.g., number or title);
- Risk description;
- Probability;
- Effect;
- Level;
- Test priority;
- Mitigation action;
- Dependencies and assumptions.

3.3.2.2.2 Perception of Risks

The performance of risk analysis is more or less subjective. In fact, most *risk analysis is based on perceptions;* it is usually not possible to determine risk probability and effect totally objectively. There is an element of prediction in risk analysis since we have to deal with something that has not happened and maybe never will.

Perceptions are personal and different people have different "pain thresholds."

Just look around you: Some people use their holidays to explore new places, others always spend their holidays at the same place. In connection with process improvements, we sometimes say that if it blows hard some people build shelters others build windmills.

People in different professions may also have different viewpoints on risks—partly because people choose jobs according to their personalities, partly because job-related experiences influence their perception of different risks.

The following descriptions of job-related risk perceptions are of course gross generalizations, but they can be used as guidelines in understanding different viewpoints on "the same risks." The descriptions encompass the following roles:

- Project managers;
- Developers;

- Testers;
- End users.

Project managers are often under time pressures, and they are used to compromises. They know that even though things may look dark, the world usually keeps standing.

Developers (i.e., analysts, designers, and programmers) are proud of their work, and they know how it was done. They have really done their best, and they are usually reluctant to accept that defects may still remain.

Testers often have a pessimistic view about work products, product components, and products. We remember previous experiences where we received objects for test and got far more failures than we expected.

The *end users* are, despite what we might think, usually highly failure tolerant. They also tend to remember previous experiences, but what they remember is that, even though the system failed, they found a way around it or another way of doing their work. End users use our product as a tool in their job, and nothing more. If it does not help them, they'll find another way of using the tool, another tool, or just live with it.

In risk analysis we must encourage communication and understanding among stakeholders. *Stakeholders need to be able to, if not agree with then at least be aware of and accept others' points of view.* If need be, stakeholders will have to compromise or use composite analysis, as explained below.

3.3.2.2.3 Scales for Risk Analysis

The analysis of risks uses metrics for probability and effect. For all work with metrics, it is mandatory to use agreed-on and understood scales.

We can work with two different kinds of scales: qualitative and quantitative. In a qualitative scale we work with feelings or assessments.

For effect we could use:

bad—worse—worst

For probability we could use:

not likely—likely—very likely

In a quantitative scale, on the other hand, we work with exact measures or numbers.

For effect we could use actual cost in for example $ or €.

Probability could be expressed as:

<= 10%, >10% & <= 50%, >50% & <=80%, > 80%

Whichever way we do it, the test manager must define scales for both probability and effect before we start the risk analysis (that is, before we assign values to the probability of the identified risks actually materializing, and values for the effect if they do).

3.3.2.2.4 Effect

The effect is the impact or consequences of a risk if it occurs. As mentioned above, the first thing we must have is a scale for the effect.

The obvious *quantitative scale* for the effect is the actual cost imparted by a risk occurring. The actual cost can be measured in any agreed-on currency (e.g., $ or €). This is an open scale; in theory, there is no limit to actual cost.

It can be very difficult to assess what the actual cost in real money might be. On the other hand, it can be quite an eye-opener to consider all the sources of extra costs associated with a risk.

Costs may be considered for:

- Time for the end user to realize that something is wrong;
- Time to report the incident to first-level support;
- Time for first-level support to understand the report and try to help;
- Time for any double or extra work to be performed by the end users;
- Loss of production because the system is down or malfunctioning;
- Time for escalation to second-level support;
- Time for second-level support to try to help;
- Time to investigate the failure and decide what to do about it;
- Time for finding the defect(s);
- Time for corrections to be implemented and tested in all affected objects;
- Time for retest and regression test;
- Time to reinstall the new version;
- Time to update what has been done by other means while the system was unavailable or malfunctioning.

These are all examples of time to be spent in connection with a failure. Costs may also be associated with, for example, renting or replacing parts of the system or the entire system.

Furthermore, there may be an effect in the form of indirect losses from people getting hurt, the environment being destroyed, or the company getting an adverse reputation or losing trustworthiness.

Failures have been known to cost lives or to put companies out of the market completely. Fortunately, it is usually not that bad, but still the effects of failures can be significant.

Another way to measure effect is by using a *qualitative scale*. Such a scale could be expressed as shown in the table below.

Effect	Description	Score
Critical	Goals cannot be achieved	6
High	Goals will be jeopardized	5
Above middle	Goals will be significantly affected	4
Below middle	Goals will be affected	3
Low	Goals will be slightly affected	2
Negligible	Goals will be barely noticeably affected	1

Inspired by Paul Gerrard

In the table there is a column for a mnemonic for the effect, a column describing the effect more precisely, and a column for a score.

Use of a numeric score makes it possible to calculate the risk level even when a qualitative scale is used for the effect.

It can be useful to define a scale with an even number of scores. This can mitigate the effect of some people having a tendency to choose the middle value if they are not sure what to score or can't be bothered to think deeper about their opinion. A scale with an even number of scores does not have a middle value, and the stakeholders will have to decide if they want to score over the middle or under.

The important point before the analysis of the effects can start is that *the stakeholders agree to and understand the scale.*

When you perform the analysis of the effect of the risks you have identified, you must keep your focus on the effect. You must *NEVER let the probability influence the effect!* It can sometimes be tempting to give the effect an extra little turn upward if we know (or think) that the probability of the risk materializing is high, but this will give a twisted picture of the risk level and should be avoided.

A simple effect analysis for the risks pertaining to the four top-level architectural areas defined for a product may look like this, using a scale from 1 to 6 where 6 is the worst

The product in the example is from the case story.

Risk Area	Effect
Setup	2
Conveyor	2
Concentration calculation	6
Compound determination	5

Often it is not enough to have one single score for the effect. Stakeholders see the effect from different perspectives. An end user sees the effect in the light of how a failure will influence his or her daily work. A customer

may look at the effect of failures on the overall business goals. A supplier organization may assess the effect in terms of correction efforts for failures or loss of credibility in the market.

These different perspectives can be honored if we use a more complex or composite effect analysis. The scale should be the same for all the perspectives, but the descriptions should be tailored to make sense for each of the viewpoints.

A composite effect analysis taking more perspectives into account may look like this:

Risk Area	Effect for Perspective			Final Effect
	User	Customer	Supplier	
Setup	5	3	2	3.3
Conveyor	3	3	5	3.7
Concentration calculation	2	5	2	3
Compound determination	1	5	3	3

Here all perspectives have the same weight, and the final effect is a simple average of the effect contributions.

If the scale is not sufficiently differentiated, the individual perspectives may be assigned independent weights, and the final effect can then be calculated as the weighted average:

Final effect = $\sum$(effect*weight)/$\sum$(weight)

The effect analysis taking more perspectives into account and assigning different weights to the perspectives may look like that shown below.

Risk Area	Effect for Perspective			Final Effect
	User W=2	Customer W=7	Supplier W=1	
Setup	5	3	2	3.3
Conveyor	3	3	5	3.2
Concentration calculation	2	5	2	4.1
Compound determination	1	5	3	3.9

3.3.2.2.5 Probability

The probability is the likelihood of the materialization of a risk.

Also here we first of all need to agree on a scale. On a *quantitative scale* probability can be measured on a scale from 0 to 1 or a scale from 0% to 100%. For most risks it is, however, almost impossible to determine the probability with such precision.

A *qualitative scale* for probability is usually much more useful, and it could be expressed as shown in the table below where there is a column for probability intervals, a column describing the probability, and a column for the score. Again using numeric scores makes it possible to calculate the risk level even when a qualitative scale is used for the probability.

Probability (%)	Description	Score
99–90	Highly likely	6
89–70	Likely	5
69–50	Above 50–50	4
49–30	Below 50–50	3
29–10	Unlikely	2
9–1	Highly unlikely	1

Focus must be kept on the probability when you perform the analysis of the probability of the risks you have identified. You must *NEVER let the effect influence the probability!* It can sometimes be tempting to give the probability an extra little turn upward if we know (or think) that the effect of the risk if it materializes is high. This will give an untrue picture of the risk level and should be avoided.

The probability of a risk materializing may be a function of many factors, for example:

- Complexity of the product or the code;
- Size of the product or the code;
- The producer of the work product(s) or component(s);
- Whether it is a new product or code or maintenance;
- The previous defect record for the product or area;
- The developers' familiarity with tools and processes.

Just like for the effect, the final probability can be calculated as the weighted average of the probabilities pertaining to the different factors.

Final probability = Σ(probability*weight)/Σ(weight)

A composite probability analysis may look like this:

Risk Area	Probability for Factor			Final Probability
	Size W=1	History W=5	Complexity W=2	
Setup	4	2	1	2
Conveyor	5	3	5	3.8
Concentration calculation	3	1	2	1.5
Compound determination	3	1	5	2.2

The same quantitative scale must be used for all factors.

3.3.2.2.6 Risk Level

The risk level is calculated for each of the identified risks as

Risk level = final effect × final probability

Using the above examples for final effect and final probability, the final risk level may look like that shown in the table below.

Risk Area	Final Effect	Final Probability	Final Risk Level
Setup	3.3	2	6.6
Conveyor	3.2	3.8	12.2
Concentration calculation	4.1	1.5	6.2
Compound determination	3.9	2.2	8.6

Sometimes some stakeholders will be unhappy with the final level. If a stakeholder has high rates for a particular risk and the risk comes out with a relatively low final risk level, this can "seem unfair." In such cases the perspectives and the scales will have to be discussed once more.

The point of the perspectives and the scales is that they should satisfy every stakeholder's viewpoint. If that is not the case, they should be adjusted. Most of the time, however, stakeholders recognize that the perspectives and scales are OK, and that their viewpoint is fairly overruled by others' different viewpoints.

The distribution of the final risk level over individual risks is used to plan the test activities. It can be used to prioritize the test activities and to distribute the available time and other resources according to the relative risk level. A test plan based on a risk analysis is more trusted than a plan based on "gut feeling."

It is usually not a very difficult task to perform a risk analysis as explained above. A full analysis, including identifying about 30 risks and assessing and calculating the effect, probability, and final level, can be done in a couple of hours. It is well worth the effort because it gives everybody involved a much clearer picture of why testing is necessary and how it should be planned.

It is difficult to predict events, and therefore all risk analysis has some built-in uncertainty. *A risk analysis must be repeated at regular intervals as the testing progresses.*

The test results can be used as input to the continuous risk analysis. If we find more defects than expected in a particular area, it means that the probability is higher than we expected, and the area hence has a higher risk level. On the other hand, if we find fewer defects than expected, the risk level is lower.

3.3.3 Risk Mitigation

We use the results of the risk analysis as the basis for risk mitigation, the last activity in risk management. "To mitigate" means "to make or become milder, less severe or less painful." That is what we'll try to do.

Faced with the list of risks and their individual risk level we have to go through each of the risks and decide

- What are we going to do?
- How are we going to do it?
- When are we going to do it?

3.3.3.1 What to Do to Mitigate Risks

In terms of what to do, we have the following choices:

- Do nothing;
- Share the pain;
- Plan contingency action(s);
- Take preventive action.

We can choose to do nothing if the benefit of waiting to see how things develop is greater than the cost of doing something.

You would not buy a safe for 1,000 € to protect your jewels if they were only worth 500 € (including the sentimental value). If the jewels were stolen, you could buy new ones and still have money left.

Sharing the pain is outside the scope of testing, but it is a possibility for project management or higher management to negotiate sharing the pain of the effect of a materializing risk with other parties. This other party could be an insurance company or it could be a supplier or even the customer.

Planning contingency action(s) is a natural part of most risk mitigation. The contingency action is what we are going to do to mitigate the effect of a risk once it actually has materialized. For risk types others than product risks and processes other than test, the response to the risk analysis may be the production of contingency plans. But it is not something that is applicable in the test planning for mitigating product risks.

Extra training is planned if it turns out that the system is more difficult to learn than expected.

Testing is one of a number of possible preventive actions. The aim is to mitigate the risks. Testing can be used to mitigate the risk level by lowering or eliminating the probability of the risk.

Product risks are associated with the presence of defects. The effect is associated with the failure of the product in use. The probability is associated with the probability of undetected defects still being present in the product when it is released, which then cause the product to fail in use.

Testing aims at identifying defects by making the product fail—before it reaches the customer. Defects found in testing can be corrected—before the system reaches the customer. Hence testing and defect correction (!) reduce the risk level by reducing the probability.

3.3.3.2 How to Mitigate Risks by Testing

When we have decided to do something to mitigate a risk, we must determine what to do. The nature of the risk can be used to determine what testing to perform to mitigate the risk, and the level of formality applied. The decisions must be documented in the applicable test strategy or test plan.

Certain test subprocesses are especially applicable for certain types of risks. We need to look at the risk source and determine the activities that are most likely to unveil defects.

Some examples are:

Risk Source	Recommended Test Phases
High risk of defects in algorithms	–Review of detailed design –Review of code –Component testing –Functional system test
Risk of problems in the user interface	–Usability evaluation of prototype (requirements review) –Usability test (nonfunctional system test)
Risk of performance problems	–Review of detailed design –Review of code –Performance test (nonfunctional system test)
Risk concerning external interface	–Review of design –Review of code –System integration test

The formality of the test can also be determined from the risk level. The rule is simple:

The higher the risk level => The higher the formality

The formality can change from test subprocess to test subprocess and it can change over the product. Some areas can have more formal testing than others, even within the same test subprocess.

In a test subprocess, for example, a system test, we can have different levels of formality as shown in the following table.

System Test	
Risk Level	**Recommended Test Phases**
High	–Specific test case design techniques to be used –Strict test completion criteria –Strict regression test procedures
Low	–Free choice of test case design techniques –And/or less strict test completion criteria –And/or less strict regression test procedures

3.3.3.3 When to Mitigate Risks by Testing

We can use the results of the risk analysis to prioritize the test activities that we have identified for the risks, and to distribute the test time (and possibly other resources).

In the prioritization we are going to determine the order in which to attack the risks. Even if we are not going to perform all of the testing activities identified for the risks in strictly sequential order, it helps in the planning stage to have them prioritized.

The priority can follow the final risk level. This means that the final level can be used as the sorting criteria. This takes every perspective of the risks into consideration in one go.

Ex.

With the example from above, the priority of the risks areas can then be as shown below, where 1 is the highest.

Risk Area	Final Risk Level	Priority
Setup	6.6	3
Conveyor	12.2	1
Concentration calculation	6.2	4
Compound determination	8.6	2

The stakeholders could also choose to let the prioritization be guided by either the final effect or the final probability. Or they may even agree to use one particular perspective, for example, the probability related to the complexity, from which to prioritize.

To calculate the distribution of the time to spend on the testing, we need to calculate the sum of the final level.

Ex.

Risk Area	Final Level
Setup	6.6
Conveyor	12.2
Concentration calculation	6.2
Compound determination	8.6
Total	33.6

The next step is to calculate the distribution of the final levels over the risk areas. This could look as shown here, where the percentages have been rounded to the nearest whole number.

Risk Area	Final Level	% Distribution
Setup	6.6	20
Conveyor	12.2	36
Concentration calculation	6.2	18
Compound determination	8.6	26
Total	**33.6**	**100%**

With a table like this we have a strong planning tool. No matter which resources we have at our disposal, we can distribute them on the risk areas and hence ensure that each area is indeed tested, but neither more nor less than it deserves.

If the project manager allocates, for example, 400 hours for the complete testing of our sample system, we can distribute this time over the areas as shown here:

Risk Area	% Distribution	Hours for Testing
Setup	20	80
Conveyor	36	144
Concentration calculation	18	72
Compound determination	26	104
Total	**100%**	**400**

The list of prioritized risks with their allocated resources and identified testing activities allows us to produce a substantiated plan and schedule for the test.

The list also allows us to immediately assess the results of a proposed change in resource allocation. If the resources we have distributed are cut, we will have to find out how to make do with what is left.

Usually we are operating with time; usually a certain number of hours are allocated for the testing and consequently a number of hours may need to be cut. If time is cut we must ask management what to do with our distribution of time over the risks. We can't leave our plan and schedule untouched—the cut must have an effect. What we can do is in principle:

- Reduce testing time proportionally to the cut.
- Take risk areas out of the testing completely.

The best thing to do is to reduce the time proportionally. This ensures that all areas are still tested, that is, that we will get some information relating to all risks. We can combine the two approaches, but we should be

very careful if we take areas that are still in the scope of the project out completely.

If some testing has already been performed, a renewed risk analysis is necessary before we can act on any cuts. In this case we must distribute the remaining resources over the remaining risks according the new final level, and prioritize as we did before.

3.4 Test Estimation

3.4.1 General Estimation Principles

This section will concentrate on time estimation, but some of the techniques may be applied for estimating elements other than time, for example, number of test cases, number of defects to be found, and number of iterations in the test process needed to fulfill the completion criteria. We may also estimate other costs, such as acquisitions of hardware, tools, etc.

Time estimation is a prediction of how much time it takes to perform an activity. It is an approximation or judgment, not a precise calculation.

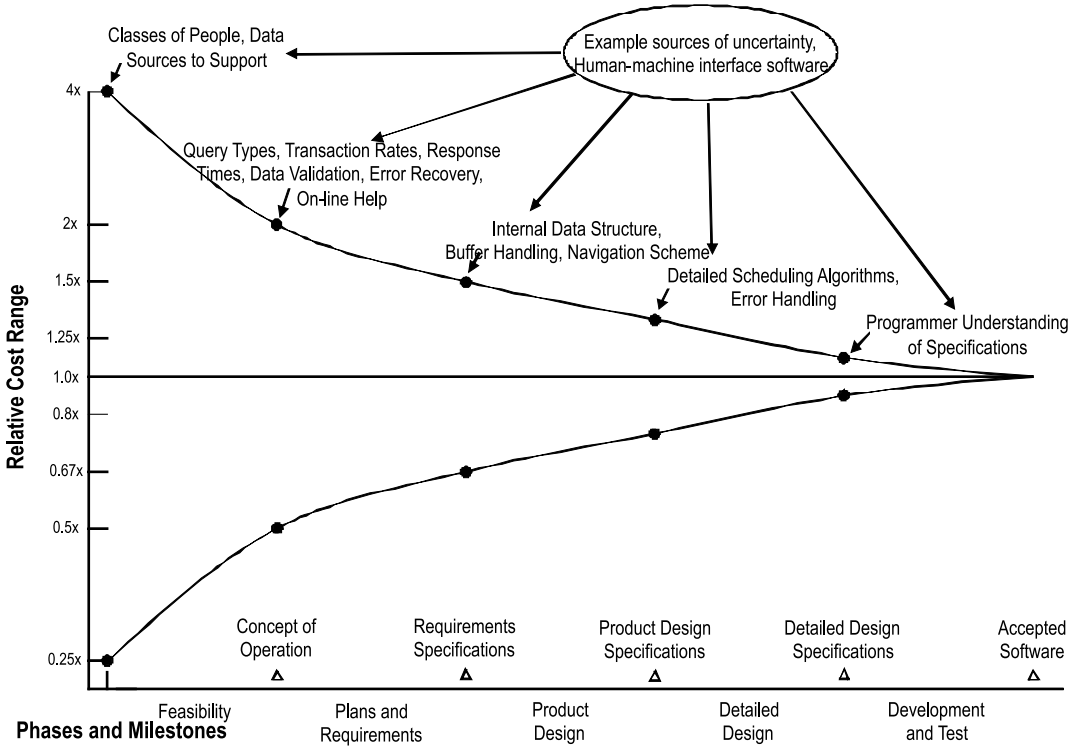
There are many ways in which to express estimations, but the best way is in hours. By using hours, we don't have problems with holidays, effective working hours, etc. You must never express estimates using dates—dates and estimates are incompatible. Estimation is input to the scheduling. Only in that activity will we transform the estimated hours into dates.

We should always take our estimations seriously. Be honest when you estimate—even though it is often easier to ask for forgiveness rather than permission. Keep your original estimates for future reference.

In line with this remember that estimation is **not**:

- The most optimistic prediction you can think of;
- Equal to the last estimate that was made;
- Equal to the last estimate + the delay the customer or the boss is willing to accept;
- Equal to a given "correct" answer.

Estimates are predictions about the future and predictions are by definition uncertain. The closer we come to the actual result, the less the uncertainty, as illustrated below.



You should always estimate the uncertainty for an estimate and document this uncertainty with the estimate.

Furthermore, estimates should always be accompanied by the rationale or justification for the estimation values along with any assumptions.

3.4.2 Test Estimation Principles

Estimating test activities is in many ways like all other estimation in a project. We need to take all tasks, even the smallest and seemingly insignificant, into account.

The time to complete must be estimated for each task defined in the task section of the test plan, including all the test process activities from test planning to checking for completion.

Even though estimation of testing tasks is in many ways identical to the estimation for any other process, there are also important differences.

The test estimation is different from other project estimation, because the number of failures we will encounter is not known in advance—although it can be estimated as well.

The number of necessary iterations before the completion criteria are met is usually not known either. As a rule of thumb, at least three iterations must be reckoned with—one is unlikely to be enough, unless the completion criterion is a simple execution of all test cases, and independent on the number of outstanding defects and coverage.

Nevertheless, we have to do our best, and the estimation must also include:

- Time to report failures (incident registration);
- Possible time to wait for defect analysis;
- Possible time to wait for defect correction;
- Time for retesting and regression testing (minimum of three iterations!).

The reason why we have to cater to the need for several iterations is that, well, "*Errare humanum est!*" When we report incidents and the underlying defects are corrected by development or support staff, experience shows that not all defects are actually corrected; not because the developers couldn't be bothered, but because, for example, they can't find the defect or can only solve part of the problem. Furthermore, defect correction introduces new defects, and defect correction unveils existing defects that we could not see before.

Experience in the testing business shows that *50% of the original number of defects remain after correction.* These are distributed as follows:

Remaining defects after correction: 20%;
Unveiled defects after correction: 10%;
New defects after correction: 20%.

So if we report 100 defects, we have 20 + 20 + 10 = 50 defects to report in the next iteration, 10 + 10 + 5 = 25 defects in the third, and 5 + 5 + 2 = 12 in the fourth. This means that we will have reported and fixed 152 in all if have fixed everything after the fourth iteration.

These are general experience numbers. It is important that you collect your own measurements!

3.4.3 The Estimation Process

Estimation is a process like anything else we do. You should, of course, use your organization's standard process for estimation, if there is one. Otherwise, you can adapt an estimation procedure like the generic one described here.

1. *Define the purpose of the estimation.* Is this estimation the first approach, for a proposal, or for detailed planning?
2. *Plan the estimating task.* Estimation is not a 2-minute task. Set sufficient time aside for it, and include the activities in the plan.
3. *Write down the basis for the estimation.* Here the scope and the size of the work are determined, and all factors that may influence the estimates are registered. This includes factors related to the nature of the processes we are working by, the nature of the project we are working in, the people we are working with, and any risks we are facing.
4. *Break down the work.* This is the work breakdown structure (i.e., the listing of all the tasks to estimate). Do this to a level of detail related to the purpose of the estimation.
5. *Estimate.* Use more than one technique as appropriate.
6. *Compare with reality and reestimate.* This is the ongoing monitoring and control of how the work that we have estimated is actually going.

3.4.4 Estimation Techniques

The following estimation techniques are the most used and an expression of the best practice within estimation, and they are further described below.

- FIA (Finger in the Air) or Best Guess;
- Experience-Based Estimation:
  - Analogies and experts;
  - Delphi technique;
  - Three-point estimation (Successive calculation);
- Model-Based Estimation;
  - Function points;
  - Test points;
  - Percentage distribution.

3.4.4.1 Best Guess (FIA) Estimation

This technique is more or less pure guesswork, but it will always be based on some sort of experience and a number of (unconscious) assumptions. The technique is very widespread, but the estimate is usually highly uncertain it is often not repeatable, and it is not always trusted.

The uncertainty is probably around 200% to 400% for estimates based on best guess. Fortunately, we can do better than that.

3.4.4.2 Analogies and Experts Estimation

In the analogy techniques you base your estimate on something you have experienced before.

For example: "This looks very much like the system I tested in my previous job. That took us 3 months, and we were four people. This is slightly smaller and we are five people—so I guess this will take 2 months to complete."

If you have participated in a testing project that is comparable to the one you are estimating, you might use that as a baseline for your estimation.

Analogies may also be based on metrics collected from previous tests. We may estimate the number of iterations of the test based on recent records of comparable test efforts. We can calculate the average effort required per test on a previous test effort and multiply this by the number of tests estimated for this test effort.

Experts, in the estimation context, know what they are talking about and have relevant knowledge. It is almost always possible to find experts somewhere in the organization.

If experts on this kind of testing are available, then by all means make use of them. They have been there before, so they know what they are talking about.

3.4.4.3 Delphi Technique Estimation

This is a simple technique that has proved remarkably resilient even in highly complex situations.

You must appoint an estimation group as appropriate; members can be stakeholders and/or experts in the tasks to estimate.

The steps in this estimation process for a specific task are:

- Each member of the group makes an estimate.
- The group is informed about the average and distribution of the provided estimates.
- The people giving estimates in the lower quartile and in the upper quartile are asked to tell the rest of the group why their estimates were as they were.
- The group estimates again, this time taking the previous result and the provided arguments for the "extreme" estimates into account.
- This may continue two, three, four or more times until the variation in the estimates is sufficiently small.

Usually the average of the estimations does not change much, but the variation is rapidly decreased. This gives confidence in the final estimation result.

The Delphi technique can be used in many ways. The people taking part can be in the same room, but they may also be continents apart and the technique used via, for example, e-mail.

The technique can be combined with other techniques. Most often the participants give their initial estimates based on experience and/or they are experts in a specific area. The initial estimates may also be obtained using some of the other estimation techniques to make them even more trustworthy.

3.4.4.4 Three-Point Estimation

Three-point estimation is a statistical calculation of the probability of finishing within a given time. The technique is useful for quantifying uncertainty to the estimate. The technique is also called successive calculation because tasks are broken down and the estimates successively calculated until the variance is within acceptable limits.

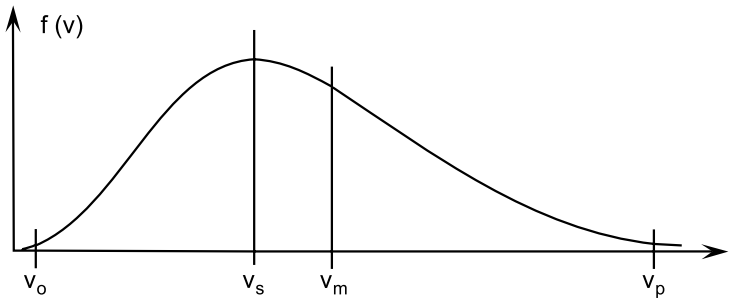
Three-point estimation is based on three estimates:

- The most optimistic time (ideal conditions);
- The most likely time (if we do business as usual);
- The most pessimistic time (Murphy is with us all the way).

The three estimates to be used can be provided in a number of ways, usually using another estimation technique high and low values may either be estimated separately (i.e., "What are the best and the worst cases?") or they may be the highest and the lowest of the individual estimates.

From the three estimates, "best, worst, and most likely," it is possible to define the distribution function for the time to finish.

It could look like shown here, where V_o = most optimistic; V_s = most likely; V_p = most pessimistic; and V_m = mean.



We can use the approximated formula to derive:

$V_m = (V_o + 3*V_s + V_p)/5$ (weighted mean)

$S = (V_p - V_o)/5$ (the standard deviation)

Based on this distribution we can calculate the time needed for any probability of finishing the task we want by using the appropriate formula.

For the 5% interval the formulas are:

5%: $V_m - 2S$ 95%: $V_m + 2S$

Let us say that for a testing task we have reckoned:

$V_o = 70$ hours $V_s = 80$ hours $V_p = 110$ hours

We calculate:

$V_m = (V_o + 3*V_s + V_p)/5 = (70 + 3*80 + 110)/5 = 84$
$S = (V_p - V_o)/5 = (110 - 70)/5 = 8$

The upper value in the 95% interval = 84 + 2*8 = 100.

Therefore if we want to be 95% sure that we'll finish in time, our estimate for the given task should be 100 hours.

All tasks or a selection of the most critical tasks can be estimated using this technique.

Tools are available to support the calculations.

3.4.4.5 Function Point Estimation

This technique is a factor estimation technique initially published by Albrecht in 1979. It has been revised several times, and it now maintained by the International Function Points User Group (IFPUG). The group has been permanent since 1992, and it is still going strong issuing new versions of the technique.

The estimation is based on a model of the product, for example a requirements specification and/or a prototype. Five aspects of the product are counted from the model:

- External inputs;
- External outputs;
- External enquiries;

- Internal logical files;
- External interface files.

The counts are then multiplied with a weight, and the total of the weighted counts is the unadjusted sum. The actual effort in person-hours is then calculated with an adjustment factor obtained from previous project data.

It requires some training to be able to count function points correctly. Continuous comparisons of actual time spent with the estimates are essential to get the best possible local adjustment factor.

The disadvantage of using function points is that they require detailed requirements or other fairly detailed descriptions of the expectations.

3.4.4.6 Test Point Estimation

In 1999 Martin Pol et al. published a dedicated test estimation technique called test points as part of the TMAP method.

The technique is based on the function point technique, and it provides a unit of measurement for the size of the high-level test (system and acceptance tests) to be executed.

The technique converts function points into test points based on the impact of specific factors that affect tests, like:

- Quality requirements;
- The system's size and complexity;
- The quality of the test basis (the document(s) the test is specified toward);
- The extent to which test tools are used.

3.4.4.7 Percentage Distribution Estimation

Unlike all the other techniques discussed here, this technique is a so-called top-down estimation technique. The fundamental idea is that test efforts can be derived from the development effort.

The estimation using this technique is starting from an estimate of the total effort for a project. This estimate may be the result of the usage of appropriate estimation techniques at the project management level, or it may be a period of time that is fixed for some reason.

The next step is to use formulas (usually just percentages) to distribute this total effort over defined tasks, including the testing tasks. The formulas are based on empirical data, and they vary widely from organization to organization.

It is essential that you get your own empirical data and constantly update them according to experiences gained.

If you do not have any data, you could assume that the total testing effort is 25% to 30% of the total project effort. The testing effort should then be spread out on the test subprocess with an appropriate amount for each test subprocess.

This example is from Capers Jones *Applied Software Measurements*. It is for in-house development of administrative systems. The left-hand table shows the distribution of the total effort on overall task, including all testing as one task only. The right-hand table shows the distribution of the effort on detailed testing tasks (the terminology is Capers Jones'.)

Activity	**%**
Requirement	9.5
Design	15.5
Coding	20
Testing (all test phases)	27
Project management	13
Quality assurance	0
Configuration management	3
Documentation	9
Installation and training	3

All phases	**%**
Component testing	16
Independent testing	84
	100
Independent testing	**%**
Integration testing	24
System testing	52
Acceptance testing	24
	100
System testing	**%**
Functional system testing	65
Non-functional system testing	35
	100

3.4.5 From Estimations to Plan and Back Again

The estimation is done to provide input to the scheduling activity in the project planning.

In the scheduling we bring the estimates for the defined testing tasks together with the people who are going to be performing the tasks. Based on the start date for the first task and the dependencies between the tasks, we can then puzzle the tasks together and calculate the expected finishing date.

As mentioned above, time estimations should be given in hours. The scheduling provides the dates: dates for when the performance of each of the tasks should begin, and dates for when they are expected to be finished.

When defining the expected finish date for a task, we need to take several aspects into account:

- The start and/or finish dates of others tasks that this task depends on to start, if any;
- The earliest possible start date for the task;

- The general calendar regarding public holidays;
- The pure estimate for the time to finish the task;
- The efficiency of the employee(s) to perform the task—typically 70% to 80% for a full-time assignment;
- The employee availability on the task (which should NOT be less than 25%).

We should not expect that our estimations will be accepted straightaway. Making plans for a development project is a very delicate balance between resources (including costs), time, and quality of the work to be done. Testing is often on the critical path for a project, and testing estimates are likely to be the subject of negotiations between stakeholders—typically the customer or higher management, the project manager, and the test manager.

The estimating does not stop with the preparation of the first schedule. Once the actual testing has started—from the first planning activities onward—we need to frequently monitor how realities correspond to the estimates. Based on the new information gathered through the monitoring, we must reestimate when the deviations between estimates and reality get too large to stay in control. Only when all of the testing activities have been completed can we stop the monitoring and reestimation.

3.4.6 Get Your Own Measurements

All estimates are based on experience, whether they are very informal (like FIA) or very formal (like function points). The better the basis for the estimation is, the better the estimation gets. Better estimation means more reliable estimations, and that is what we both, management and customers, want.

To get better estimates, we need to collect actual data related to the estimates. The more empirical data we have, the better future estimates will become. In general, we can say that (almost) any empirical data is better than no data.

We do however always need to objectively evaluate the empirical data we have—are they collected from tasks that can be compared with the ones we are dealing with now?

When we use the available empirical data, we also have an obligation to contribute to and refine these empirical data on an ongoing basis. Empirical data for estimation is part of the measurements we are collecting. So we need to chip in to establish a set of simple measurements of time, costs, and size for all projects in which we participate.

This is described in the following section.

3.5 Test Metrics and Measurements

Tom De Marco, one of the testing gurus, once said:

> *If you don't measure you're left with only one reason to believe you're in control—hysterical optimism.*

One of the principles of good planning, both of testing and anything else, is to define specific and measurable goals for the activities. But it is not enough for goals to be measurable; we must also collect facts—measurements—that can tell us if we have indeed achieved the goals.

3.5.1 Measuring in General

For facts or data collection we operate with the following concepts:

- Metric—a definition of what to measure. The definition must include data type, scale, and unit.
- Measuring method—the description of how we are going to get the data.
- Measurements—the actual values collected for the metrics.

An example could be that the metric for the size of a book is "number of pages"; the measuring method is to "look at the last page number"; and the measurement for Alice in Wonderland, ISBN 7409746, is "54."

It is a good idea to establish a measurement plan as part of the project plan or master test plan. This should specify the metrics we want to measure and the measuring methods; who is going to measure; and, perhaps most importantly, how the measurements will be analyzed and used.

Our measurements are derived from raw data such as time sheets, incident reports, test logs, and work sheets. *Direct measurements* are measurements we get directly from the raw data, for example, by counting the number of log sheets for passed test procedures and counting the number of incident reports. *Indirect measurements* are measurements we can calculate from direct measurements.

Most direct measurements have no meaning unless they are placed in relation to something. Number of incidents as such—for example, 50—says nothing about the product or the processes. But if we calculate the defects found compared to the estimated amount of defects it gives a much better indication—either of our estimation or of the quality of the product!

It is a common mistake to think that only objective data should be used. Objective data are what you can measure independently of human

opinions. But even though subjective data have an element of uncertainty about them, they can be very valuable. Often subjective data are even cheaper to collect than objective data.

A subjective metric could be: The opinion of the participants in walk-throughs concerning the usefulness of the walk-through activity on a scale from 1 to 5, where 1 is lowest and 5 is highest. This is easy to collect and handle, and it gives a good indication of the perception of the usefulness of walk-throughs.

The metrics should be specified based on the goals we have set and other questions we would like to get answers to, such as how far we have got in performing a specific task in relation to the plan.

3.5.2 Test-Related Metrics

Many, many measurements can be collected during the execution of test procedures (and any other process for that matter). They can be divided into groups according to the possibilities they provide for control. The groups and a few examples of direct measurements are listed here for inspirational purposes; the lists are by no means exhaustive.

Measures About Progress:

- Of test planning and monitoring:
  - tasks commenced;
  - task completed.
- Of test development:
  - number of specified test procedures;
  - number of approved test procedures;
  - relevant degrees of coverage achieved in the specification, for example for code structures, requirements, risks and/or business processes;
  - other tasks completed.
- Of test execution and reporting:
  - number of executed test procedures (or initiated test procedures);
  - number of passed test procedures;
  - number of passed retests;
  - number of test procedures executed for regression testing;
  - other tasks completed.
- Of test closure:
  - tasks completed.

For each of these groups we can collect measurements for:

- Time spent on specific tasks both in actual working hours and elapsed time;
- Cost both from time spent and from direct cost such as license fees.

Measures About Coverage:

- Number of coverage elements covered by the executed test procedures (e.g., code structures, requirements, decision combinations, classification tree leaves).

Measures About Incidents:

- Number of reported incidents;
- Number of incidents of different classes (e.g., defects, misunderstandings, or enhancement requests);
- Number of defects reported to have been corrected;
- Number of closed incident reports.

Measures About Confidence:

- Subjective statements about confidence from different stakeholders.

All of these measurements should be collected at specific points in time, and the time of the measuring should be noted to enable follow-up on the development over time, for example, the development in number of open incident reports on a weekly basis.

It is also important to prepare to be able to measure and report status and progress of tasks in relations to milestones defined in the development model we are following.

To be able to see the development of measured topics in relation to expectations, corresponding factual and/or estimated total numbers are also needed. A few examples are:

- Total number of defined test procedures;
- Total number of coverage elements;
- Total number of failures and defects;
- Actual test object attributes (e.g., size and complexity);
- Planned duration and effort for tasks;
- Planned cost of performing tasks.

3.5.3 Analysis and Presentation of Measurements

It is not enough to just collect measurements; they must be analyzed and presented in an understandable way to relevant stakeholders to be of real value.

The analysis and presentation of measurements are discussed in Section 3.6 below.

3.5.4 Planning Measuring

It is important that stakeholders agree to the definition of the metrics and measuring methods before any measurements are collected. Unpopular or adverse measurements may cause friction, especially if definitions are not clear and approved. You can obtain very weird behaviors by introducing measures!

Here is some advice you should keep in mind when you plan the measures you are going to collect. You need to aim for:

- *Agreed metrics*—definitions (e.g., what is a line of code), scale (e.g., is 1 highest or lowest?), and units (e.g., seconds or hours) must be agreed and understood.
- *Needed measures*—what is it you want to know, to monitor, and to control?
- *Precise measures*—appropriate scale must be used.
- *Comparable measures*—for example, over time or between sources.
- *Economical measures*—practical to collect and analyze compared to the value of the analysis results.
- *Creating confidentiality*—never use measurements to punish or reward individuals.
- *Using already existing measurements*—maybe the measurements just need to be analyzed in a new way.
- *Having a measurement plan*—the plan should outline what, by whom, when, how, and why the measures are going to be collected.
- *Using the measurements*—only measure what can be used immediately and give quick and precise feedback.

3.6 Test Progress Monitoring and Control

Continuous monitoring on how the testing is progressing compared to the plan is absolutely necessary to stay in control. If we don't control the test project, it will control us—and that is not a nice experience.

3.6.1 Collecting Data

The data to collect during testing should be specified in the "Metrics to be collected" section in the test plan, based on the metrics requirements outlined in the test policy and the relevant test strategy.

Regardless of which measurements we have planned to collect, it is not enough to just collect them. They must be presented and analyzed to be of real value.

3.6.2 Presenting the Measurements

Test reports are used to communicate test progress. These reports must be tailored to the different stakeholders (see Section 3.2 above). The stakeholders for test monitoring information include the customer, the project and/or product (or higher) management, the test management, and the testers.

The customer and the management above test management need test completion reports. The test management needs information on a continuous basis to keep in control. The testers need to be kept informed about progress on a very regular basis—preferably daily when the execution activities are at their peak.

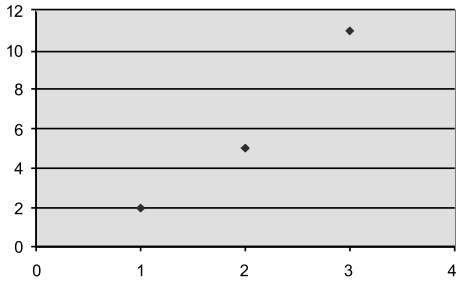
A picture speaks a thousand words. The best way to present progress information for testing is by using graphics. This holds true for all stakeholders. Graphics used in the right way give an immediate overview of the state of the testing.

The flip side of the coin is that graphics can "lie." You can do it deliberately—which is outside the scope of this book—or you can make it happen accidentally if you are too eager to make your presentation "interesting" and "lively." The truth may look boring, but adding decoration does not help.

One of the common mistakes is to use too many dimensions. Most of our information is two dimensional: the number of something at different points in time. Many graphs are, however, presenting two-dimensional information in a three-dimensional way.

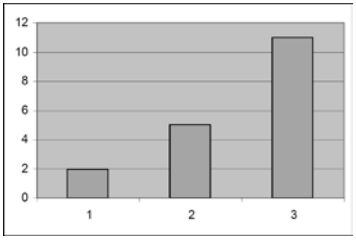
Consider the following information:

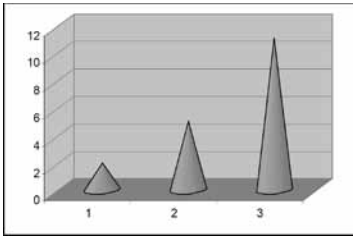
Day 1: 2 defects found
Day 2: 5 defects found
Day 3 11 defects found.



The simplest way to present this is as shown to the right of the raw data. See the trend? Yes, that is perfectly clear! Need anything else? Not really.

But all too often we may see exactly the same information presented like this: or even worse like this:





Does that add to the understanding? No. Does it blur the message? Maybe.

There is a "metric" called the ink-factor. That is defined as the amount of ink used in the presentation graph in relation to the ink needed to convey the message in a graph. You should keep the ink-factor as low as possible.

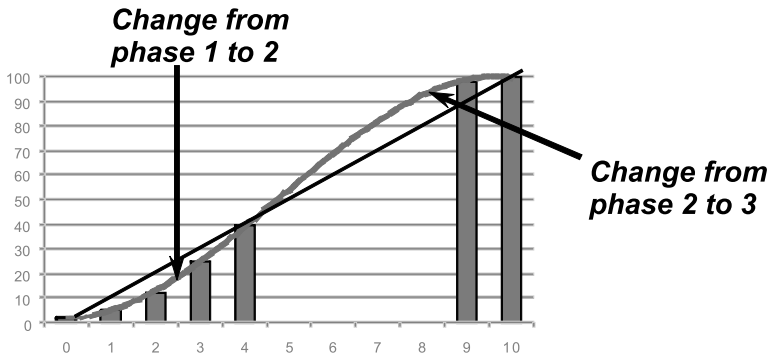
Also *avoid highlighting (read: hiding) the message in decoration, patterns, shading, or color.* A graph that presents the number of failures found each day as the size of the corollas of a line of flowers is perhaps cute, but not very professional.

More obvious ways to misinform is by changing the scale across the axis, or by omitting or distorting the context of the information or the way it has been collected.

Whichever way you choose to present the information you have collected, it is your responsibility to ensure that the recipients understand and interpret the data correctly. In the following some of the most common and useful ways of presenting test progress information are described.

3.6.2.1 S-Curves

Perhaps the most used, most loved, and most useful way of presenting progress information and controlling what's happening is S-curves. They are named so because of the shape of the "perfect" curve.



Source:
Marnie Hutcheson
Unicom Seminar
Oct. 95.

The figure shows a graph where the S-curve is the thick black line swinging in an S-shape from the starting point (0,0) to the estimated end point (10,100).

S-curves can be used for many different metrics. You need two sets of metrics that are related to each other; one is typically time, and the other could be :

- Test cases (run, attempted, passed, completed);
- Incidents (encountered, fixed, retested).

S-curves can give us early warnings of something being wrong. They can also give us reassurance that (so far) things are progressing as planned.

The principle in S-curves is that our work falls in three phases:

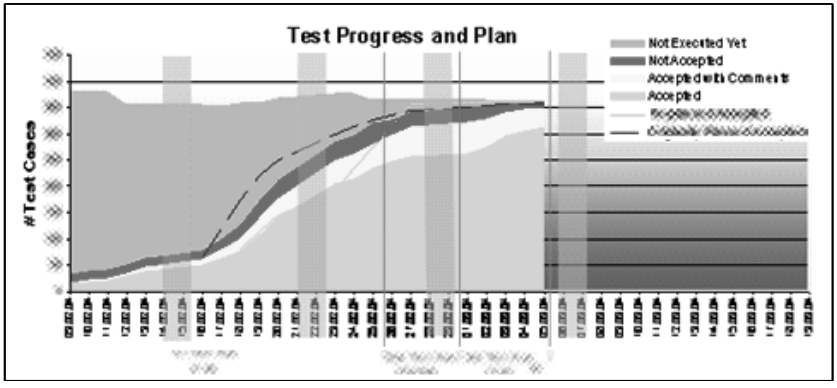
- Phase 1: slow start—not more than 15% to 25% of the time.

  An initial period of reduced efficiency to allow for testing teams to become familiar with the testing task, for example with the test object, the test environment, and execution and logging practices.
- Phase 2: productive phase—55% to 65% of the time.

  After the initial period, the second phase is the period of maximum efficiency.
- Phase 3: the difficult part—10% to 25% of the time.

  The final phase reflects the need to be able to ease off the work as the testing window nears completion.

The figure below shows how real data are reported as several S-curves in the same graph.



To use an S-curve you need to know the expected start point (usually 0,0) and the expected end point. The end point is your estimation of what needs to be achieved after a specific period of time; for example, 300 test cases passed after 21 days of testing.

You mark the start point and the end point, and you draw (or get a tool to draw) a third-order polynomial that fits.

As the time goes by and you do you work, you plot in your achievements; for example, sum of test cases passed day by day. Follow the progress to see if it fits the predicted curve. If it does, we are happy! This is illustrated in the first graph in this section.

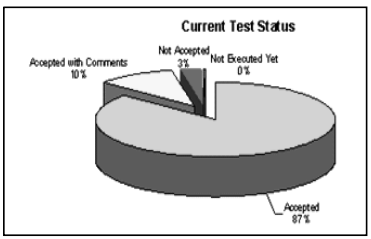
If the upward turn, marking the start of the second phase, set in too late, we are in trouble. But the good news is that we know it and can take action well before the end date! If the curve is rising too fast, we may also be in trouble, and we must investigate what could be wrong. Maybe our test cases are not giving enough failures? Maybe we have run the smallest test procedures first and are pushing work in front of us? Or maybe things are going fine and it is our estimate that is wrong.

3.6.2.2 Pie Chart

Pie charts are used to give an overview of the proportions of different aspects relative to each other.

The graph shown here gives a nice impression of the testing going well.

But think about the ink-factor—maybe the third dimension is not needed to present the information.



3.6.2.3 Check Sheets

Check sheets are a good way to give a quick overview of status. They can be used to show progress compared to plan, for example, for planned test cases, planned test areas, or defined risks.

Check sheets can be presented as colorful graphics or expressed as lists or hierarchies. They are usually easy to produce and easy to understand. Some organizations make wall-size check sheets and stick them in the meeting room or the corridor. This way everyone has easy access to the information about the progress of the testing.

A few examples of check sheets are shown below.

The first is an extract of the check sheet presented on Systematic's intranet. It is updated every day. Even though the text has been deliberately blurred and the extract is small, it gives an impression of things going well. There are no black or white fields in the status column.

Ex.

Status for project: ksdf				
Area	Remain	% Complete	Status	Comment
agfdg	8	56		
gstyk	0	100		
jl,flli	0	100		
dsrahjtdulk	1	80		
ths	2	56		
jdvw	0	100		
yjdtek	0	100		
	0	87		

Legend
- Completed
- In progress
- Blocked (see comment)
- Not started

The next is a dashboard suggested by James Bach.

Ex.

Area	Test Effort	Coverage planned	Coverage achieved	Quality	Comments
Startup	High	>80%	27%	☹	ER 52
Discount	Low	>40% <70%	53%	☺	
Pricing	**Blocked**	>40% <70%	14%	☹	**ER 86**

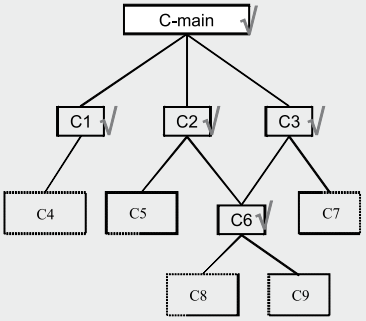
James Bach's recommendations or the presentation are to draw it on the wall, make it huge, and update it every day.

Ex.

The last example here is a tiny extract of a hierarchical check sheet showing the progress of a component test for a system.

The marked components have been successfully component tested and are ready for integration.

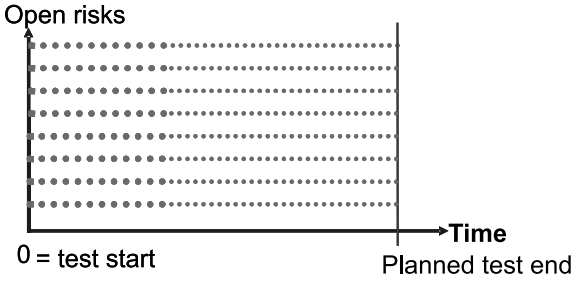
The integration testing has not yet started—no interfaces are marked as having been successfully tested. It is however easy to see which interfaces we can test now.



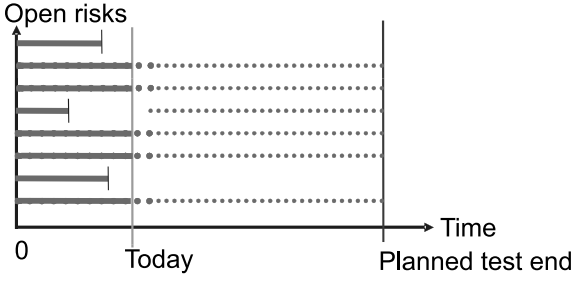
3.6.2.4 Risk-Based Reporting

If our test approach is based on identified and analyzed risks, it is appropriate to report on the test progress in terms of these risks.

The purpose of this test is to eliminate the risk, so the reporting must be in terms of eliminated risks. The open risks at the test start could be illustrated like this:



At any point in time, we must make it possible to see from the updated progress graph which risks are still open, if any. The risks with the small vertical line across them are eliminated and hence no longer present any threat to the system!



3.6.2.5 Statistical Reporting

The way a process is performed is different from project to project and over time, because processes are performed by people, not machines.

Statistics is the science of patterns in a variable world. We can say that statistics make the invisible visible. This means that statistical methods can be used to help us:

- Understand the past;
- Control the present;
- Predict the future.

Statistics also include handling of "fortuitousness," that is, happenings that are out of the ordinary.

When we have to deal with many happenings assumed to be "alike," we need to find out what "alike" means. To do that we must find out what the norm is, and what variances are allowed to still call things "normal."

Ex.

Norm and variation vary. In our family the norm is that we are friendly and talk to each other in nice, calm tones of voices. I, however, have a short temper, and sometimes raise my voice without anything being really out of the normal. If, on the other hand, I keep quiet, then my mood is not within the norm. My husband is different: if he raises his voice just a little bit, he is sure to be in a very bad mood.

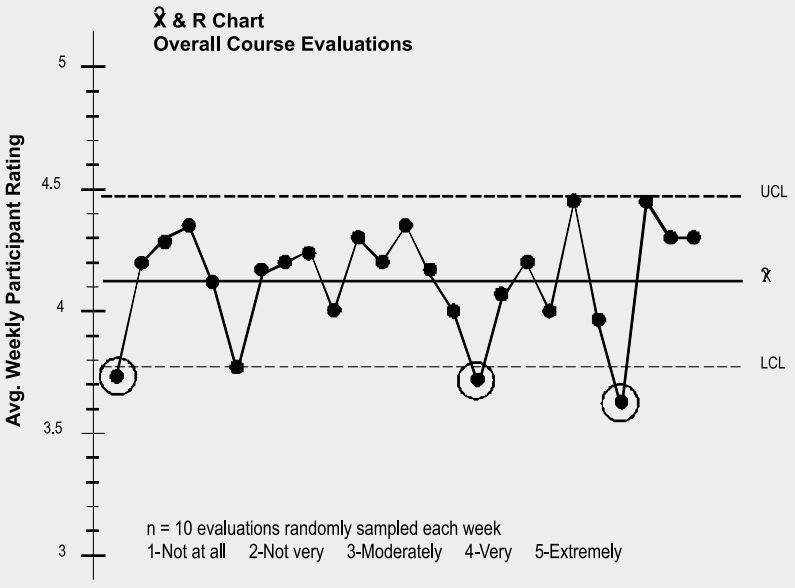
The norm in statistics can be calculated in three much used ways: the mean—the arithmetic average; the median—the value that splits the group in the middle, and the modus—the most frequent value.

But how far from the "normal" value can a given value be and still be considered within the norm? This can be calculated as the "upper control level" and the "lower control level."

All this can be illustrated in a control sheet.

Ex.

An example of a control sheet is shown here. The values are ratings for a course. Every week 10 evaluations are randomly sampled and the average is plotted in the graphs. The graph shows the upper and lower control levels (UCL and LCL, respectively) for the series of ratings.



The values here are indicators for the course performance. We can choose other values that can act as indicator values for our processes if we want to control how they are performed.

An indication value may be, for example, the average time it takes per requirement to produce test cases.

When we examine the control sheet, we should be looking for warnings of something being outside the borders of the norm. Such warnings might include:

- One value outside either control level (CL);
- Two out of three values on the same side of the mean;
- Six values in a row either up or down;
- Four values in a row alternately up and down.

Many tools can assist in the necessary statistical calculations. The use of control sheets and statistics is rather advanced process control—belonging to CMMI maturity level 4—and we will not go into further depth here.

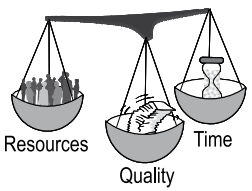
3.6.3 Stay in Control

No matter which way the progress is measured and presented, the test manager must analyze the measurements. If something seems to not be going as expected, further analysis must be made to determine what may be wrong.

Sometimes we need to take action to stay in control.

Keeping the triangle of test quality in mind, you have three aspects that can change—and at least two must be changed at a time to keep the balance. The aspects are:

- The resources available for the activity;
- The time available for the activity;
- The quality of the work to be performed.



Usually when things are getting out of control it is because we are behind schedule or because our time frame has been squeezed. To compensate for this we must try to obtain additional resources and/or change the quality of the testing. The latter can be done by changing the test completion criteria and/or changing the amount or depths of the tests to be performed.

Any change you make must be reflected in the plan. The plan must be updated with the new decisions based on the new information. The new plan must be reviewed and approved, just like the first one, and it must be communicated to all relevant stakeholders.

Remember that *it is not a virtue to comply with the plan at any cost—the virtue lies in the plan complying with reality*.

Examples of things not being as they should be are:

- The delivered software is not ready for testing.
- The easy test cases have been run first.
- The test cases are not sufficiently specified.
- The test cases do not give the right coverage.
- General defects have widespread effects.
- Defect correction is too slow.
- Defect correction is not sufficiently effective.

Based on this analysis the test manager must identify what can be done to remedy or mitigate the problems.

Possible actions might include:

- Tighten entry criteria.
- Cancel the project.
- Improve the test specifications.
- Improve the incident reporting.
- Perform a new risk analysis and replan.
- Do more regression testing.

The important message to the test manager is that he or she must intervene as soon as deviations appear!

As mentioned before:

If you do not control the test, it will control you!

To sum up, we can say that as test managers we must set out the destination and plan how to get there; collect data as we go along; analyze data to obtain information; and act on the information and change destination and plan as appropriate.

Questions

1. What are the three main activities in the test management process?
2. What does it mean that a test plan should be SMART?
3. How does a test management process interact with an IT project management process?

4. What is the relation between test project management and test subprocess management?
5. What is the relation between test subprocess management and dynamic testing?
6. How may planning be "invisible" work?
7. What are the three things that should guide monitoring?
8. How may lessons learned be collected?
9. What types of documentation are produced from a test management process?
10. What information should be present for any document?
11. What might the test item for a test be?
12. What is the relationship between the test item and features to be tested?
13. What is a stakeholder?
14. What are the two risk types that should be handled in a test plan?
15. What might test criteria include?
16. What is regression testing?
17. What might cause a pause in the test execution?
18. How can you illustrate who carries which responsibilities?
19. In which part of the test plan do the testing tasks, the estimates, and the people come together?
20. What guides when a test status report is issued?
21. What topics should be covered in a test status report?
22. What topics should be covered in a test completion report?
23. What questions can you ask to identify stakeholders?
24. What stakeholders could we have from the business?
25. What information may be collected about a stakeholder?
26. What is the purpose of a communication plan?
27. What is the golden rule of testing?
28. What is a risk?
29. What is the risk level?
30. What is a process risk?
31. What is a project risk?
32. What might a project risk be?
33. What is a product risk?
34. What type of risk is testing useful for?
35. What are the activities in risk management?
36. Which risk identification techniques could we use?
37. What is a risk checklist used for?
38. What is the main rule for brainstorming?
39. How must an expert interview be prepared?

40. What is risk analysis?
41. What information might be collected for a risk register?
42. How might the viewpoints of risk differ for different roles?
43. What kinds of scales can be used for risk analysis?
44. What is the effect of a risk?
45. What is a composite effect analysis?
46. How do you calculate the final effect?
47. What is risk probability?
48. What is the result of a risk analysis used for?
49. How many times should a risk analysis be performed and why?
50. What is mitigation?
51. What actions can be taken as a result of a risk analysis?
52. Which action is most appropriate for product risks?
53. How is formality of testing and risk levels related?
54. How can you prioritize the testing for risks?
55. How can you use risk level to distribute testing resources?
56. What can you do if testing time is cut?
57. What is estimation, and what is not estimation?
58. How precise can an estimate be?
59. How is test estimation different from ordinary project estimation?
60. What are the six steps in the estimation process?
61. What are the three estimation technique types?
62. What is the analogy technique based on?
63. What are the steps in the Delphi technique?
64. What three estimates do we need for the successive calculation estimation technique?
65. What is the estimation based on in the successive calculation estimation technique?
66. What are the five things you count for function point calculation?
67. Which test estimation technique is based on function points?
68. What is the difference between all other techniques and the percentage distribution technique?
69. What must be taken into account when defining the finish date for a task?
70. What is the difference between direct and indirect measurements?
71. How can subjective measurements be used?
72. What should be noted about the measurements we collect?
73. How can we present measurements?
74. What is the ink-factor?

75. What is the principle in S-curves?
76. What is a check sheet?
77. What is risk-based reporting?
78. Why should we use statistical reporting for process performance?
79. What must we do to stay in control?
80. What happens if we don't control the test project?

promotion

EuroSTAR Software Testing CONFERENCE | copenhagen 06-09 NOVEMBER 2017

EuroSTAR 2017 in Copenhagen (6-9 Nov) will mark our 25th anniversary – we really hope you can join us, so we're offering you **20% off** any ticket until April 19th.

Click below for your discount code.....

get discount code

EuroSTAR Software Testing

purchase

The full version of this eBook and book is now available to purchase from **www.amazon.com**

buy now

EuroSTAR Software Testing

want more?

Enjoyed this eBook and want to read more?

Check out our extensive eBook library on Huddle.

EuroSTAR Software Testing | Huddle

Join us online at the links below




EuroSTAR Software Testing

www.eurostarsoftwaretesting.com