

# Test Maturity Model Integration (TMMi)

Version 2.0

Produced by the TMMi Foundation

**Editor:** Erik van Veenendaal

Copyright Notice  
Unlimited distribution subject to Copyright  
Copyright © TMMi Foundation, Ireland.

This TMMi Foundation material is furnished on an 'as-is' basis.

The TMMi Foundation makes no warranties of any kind, either expressed or implied, as to any matter included, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. The TMMi Foundation does not make any warranty of any kind with respect to freedom from patent, trademark or copyright infringement.

Use of any trademarks in this document is not intended in any way to infringe on the rights of the trademark holder.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the TMMi Foundation.

The following registered trademarks and service marks are used in the TMMi Foundation documentation: CMM®, CMMI®, TMM<sup>SM</sup>, TMMi®, IDEAL<sup>SM</sup>, SCAMPI<sup>SM</sup>, TMap®, and TPI®.

CMM and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

IDEAL and SCAMPI are service marks of Carnegie Mellon University.

TMM is a registered service mark of Illinois Institute of Technology.

TMMi® is a registered trademark of TMMi Foundation.

TMap and TPI are registered trademarks of Sogeti, The Netherlands.

## Contributors

Laura Casci (UK)  
Vicky Chen (Canada)  
Andrew Goslin (UK)  
Murali Krishnan (India)  
Klaus Olsen (Denmark)  
Fran O'Hara (Ireland)  
Robert Magnusson (Sweden)  
Bill McGir (USA)  
Mac Miller (UK)  
Narayanamoorthy Subramanian (India)  
Alec Puype (Belgium)  
Geoff Thompson (UK)  
Tiruvallur Thattai Srivatsan (India)  
Nathan Weller (UK)  
Brian Wells (UK)

## Revisions

This section summarizes the key revisions between version 1.0 and version 2.0 of this document. This section is provided for information only.

Section	Revision Notes
2.4	Revisions to the characteristics of Level 3
3.5	Added the supporting CMMI process areas defined for the TMMi level 3 process areas
PA 2.1	Revisions to the Introductory Notes of the Test Policy and Test Strategy process area
PA 2.1	Added new specific practice SP2.1 'Perform a generic product risk assessment'
PA 2.1	Revisions and additions to typical work products, sub-practices and examples
PA 2.1	Added generic goal for GG3, correcting the omission
PA 2.2	Revisions to the Introductory Notes and Scope of the Test Planning process area
PA 2.2	Revisions and additions to typical work products, sub-practices, examples and elaborations
PA 2.3	Added Scope of the Test Monitoring & Control process area, correcting the omission
PA 2.3	Revised specific practice SP 2.2 'Monitor defects', replacing "test incidents" with "defects"
PA 2.3	Revisions and additions to typical work products and sub-practices
PA 2.4	Revisions and additions to typical work products and examples
PA 2.5	Revisions and additions to typical work products
Level 3	Added TMMi Level 3, including the five TMMi level 3 process areas: Test Organization, Test Training Program, Test Lifecycle and Integration, Non-Functional Testing and Peer Reviews
4.3	Revised mapping for TMMi Level 2 to TPI
4.4	Added mapping for TMMi Level 3 to TPI

## Contents

1	Test Maturity Model Integration (TMMi)	7
1.1	Introduction	7
1.2	Background and History	7
1.3	Sources	7
1.4	Scope of the TMMi	8
2	TMMi Maturity Levels	10
2.1	Overview	10
2.2	Level 1 Initial	11
2.3	Level 2 Managed	11
2.4	Level 3 Defined	11
2.5	Level 4 Management and Measurement	12
2.6	Level 5 Optimization	12
3	Structure of the TMMi	13
3.1	Required, Expected and Informative Components	13
3.2	Components of the TMMi	13
3.3	Generic Goals and Generic Practices	15
	GG 2 Institutionalize a Managed Process	15
	GG 3 Institutionalize a Defined Process	17
3.4	Supporting process areas for generic practices	17
3.5	Supporting CMMI process areas for TMMi	18
	TMMi Level 2: Managed	20
PA2.1	Test Policy and Strategy	21
	SG 1 Establish a Test Policy	22
	SG 2 Establish a Test Strategy	23
	SG 3 Establish Test Performance Indicators	24
	GG 2 Institutionalize a Managed Process	25
	GG 3 Institutionalize a Defined Process	27
PA 2.2	Test Planning	28
	SG 1 Perform Product Risk Assessment	29
	SG 2 Establish a Test Approach	30
	SG 3 Establish Test Estimates	32
	SG 4 Develop a Test Plan	34
	SG 5 Obtain Commitment to the Test Plan	36
	GG 2 Institutionalize a Managed Process	37
	GG 3 Institutionalize a Defined Process	39
PA 2.3	Test Monitoring and Control	40
	SG 1 Monitor Test Progress Against Plan	41
	SG 2 Monitor Product Quality Against Plan And Expectations	43
	SG 3 Manage Corrective Actions To Closure	45
	GG 2 Institutionalize a Managed Process	46
	GG 3 Institutionalize a Defined Process	49
PA 2.4	Test Design and Execution	50
	SG 1 Perform Test Analysis And Design Using Test Design Techniques	51
	SG 2 Perform Test Implementation	52
	SG 3 Perform Test Execution	54
	SG 4 Manage Test Incidents To Closure	55
	GG 2 Institutionalize a Managed Process	56
	GG 3 Institutionalize a Defined Process	58
PA 2.5	Test Environment	60
	SG 1 Develop Test Environment Requirements	61
	SG 2 Perform Test Environment Implementation	62
	SG 3 Manage And Control Test Environments	63
	GG 2 Institutionalize a Managed Process	64
	GG 3 Institutionalize a Defined Process	66
	TMMi Level 3: Defined	67
PA 3.1	Test Organization	68
	SG 1 Establish a Test Organization	69
	SG 2 Establish Test Functions For Test Specialists	70
	SG 3 Establish Test Career Paths	71
	SG 4 Determine, Plan and Implement Test Process Improvements	72

SG 5	Deploy Organizational Test Processes and Incorporate Lessons Learned .....	74
GG 2	Institutionalize a Managed Process .....	75
GG 3	Institutionalize a Defined Process .....	78
PA 3.2	Test Training Program .....	79
SG 1	Establish an Organizational Test Training Capability .....	79
SG 2	Provide Necessary Test Training .....	82
GG 2	Institutionalize a Managed Process .....	83
GG 3	Institutionalize a Defined Process .....	85
PA 3.3	Test Lifecycle and Integration .....	86
SG 1	Establish Organizational Test Process Assets .....	87
SG 2	Integrate the Test Life Cycle with the Development Models .....	90
SG 3	Establish A Master Test Plan .....	91
GG 2	Institutionalize a Managed Process .....	94
GG 3	Institutionalize a Defined Process .....	98
PA 3.4	Non-Functional Testing .....	99
SG 1	Perform a Non-Functional Product Risk Assessment .....	100
SG 2	Establish a Non-Functional Test Approach .....	101
SG 3	Perform Non-Functional Test Analysis and Design .....	102
SG 4	Perform Non-Functional Test Implementation .....	103
SG 5	Perform Non-Functional Test Execution .....	104
GG 2	Institutionalize a Managed Process .....	106
GG 3	Institutionalize a Defined Process .....	108
PA 3.5	Peer Reviews .....	109
SG 1	Establish a Peer Review Approach .....	109
SG 2	Perform Peer Reviews .....	110
GG 2	Institutionalize a Managed Process .....	112
GG 3	Institutionalize a Defined Process .....	114
4	Comparing TPI and TMMi .....	115
4.1	The TPI model .....	115
4.2	Comparing TMMi to TPI .....	118
4.3	Mapping TMMi Level 2 to TPI .....	119
4.4	Mapping TMMi Level 3 to TPI .....	122
5	Glossary .....	127
6	References .....	140

# 1 Test Maturity Model Integration (TMMi)

## 1.1 Introduction

For the past decade, the software industry has put substantial effort in improving the quality of its products. This has been a difficult job, since the size and complexity of software increases rapidly while customers and users are becoming more and more demanding. Despite encouraging results with various quality improvement approaches, the software industry is still far from zero defects. To improve product quality, the software industry has much focused on improving their development processes. A guideline that has been widely used to improve the development processes is the Capability Maturity Model. The Capability Maturity Model (CMM) and its' successor the Capability Maturity Model Integration (CMMI) are often regarded as the industry standard for software process improvement. Despite the fact that testing often accounts for at least 30-40% of the total project costs, only limited attention is given to testing in the various software process improvement models such as the CMM and the CMMI. As an answer, the testing community has created its complementary improvement models. This document describes the Test Maturity Model Integration (TMMi). The TMMi is a detailed model for test process improvement and is positioned as being complementary to the CMMI.

## 1.2 Background and History

The TMMi<sup>1</sup> framework has been developed by the TMMi Foundation as a guideline and reference framework for test process improvement and is positioned as a complementary model to the CMMI<sup>2</sup> Version 1.2 [CMMI] addressing those issues important to test managers, test engineers and software quality professionals. Testing as defined in the TMMi is applied in its broadest sense to encompass all software product quality-related activities.

**Testing:** The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects. [ISTQB]

Just like the CMMI staged representation, the TMMi also uses the concept of maturity levels for process evaluation and improvement. Furthermore process areas, goals and practices are identified. Applying the TMMi maturity criteria will improve the test process and have a positive impact on product quality, test engineering productivity, and cycle-time effort. The TMMi has been developed to support organizations at evaluating and improving their test process. Within the TMMi, testing evolves from a chaotic, ill-defined process with a lack of resources, tools and well-educated testers to a mature and controlled process that has defect prevention as its main objective.

Practical experiences are positive and show that TMMi supports the process of establishing a more effective and efficient test process. Testing becomes a profession and a fully integrated part of the development process. As stated the focus of testing changes from defect detection to defect prevention.

## 1.3 Sources

The development of the TMMi has used the TMM<sup>3</sup> framework as developed by the Illinois Institute of Technology as one of its major sources [Burnstein]. In addition to the TMM, it was largely guided by the work done on the Capability Maturity Model Integration (CMMI), a process improvement model that has widespread support in the IT industry. The CMMI has both a staged and continuous representation. Within the staged representation the CMMI architecture prescribes the stages that an organization must proceed through in an orderly fashion to improve its development process. Within the continuous representation there is no fixed level set of levels or stages to proceed through. An organization applying the continuous representation can select areas for improvement from many different categories. The TMMi has been developed as a staged model. The staged model uses predefined sets of process areas to define an improvement path for an organization. This improvement path is described by a model component called a maturity level. A maturity level is a well defined evolutionary plateau towards achieving improved organizational processes. At a later stage a continuous representation of the TMMi may become available. This will most likely not influence the content of the TMMi. It will 'only' provide a different structure and representation.

Other sources to the TMMi development include the Gelperin and Hetzel's Evolution of Testing Model [Gelperin and Hetzel], which describes the evolution of the testing process over a 40-year period, Beizer's testing model, which describes the evolution of the individual tester's thinking [Beizer], research on the TMM carried out in the EU-funded MB-TMM project, and international testing standards, e.g. IEEE 829 Standard for Software Test

<sup>1</sup> TMMi is a registered trademark of the TMMi Foundation

<sup>2</sup> CMM and CMMI are registered trademarks of Carnegie Mellon University

<sup>3</sup> TMM is a registered service mark of Illinois Institute of Technology

Documentation [IEEE 829]. The testing terminology used in the TMMi is derived from the ISTQB Standard Glossary of terms used in Software Testing [ISTQB].

As stated for defining the maturity levels, the evolutionary testing model of Gelperin and Hetzel has served as a foundation for historical-level differentiation in the TMMi. The Gelperin and Hetzel model describes phases and test goals for the 1950s through the 1990s. The initial period is described as “debugging oriented”, during which most software development organizations had not clearly differentiated between testing and debugging. Testing was an ad-hoc activity associated with debugging to remove bugs from programs. Testing has, according to Gelperin and Hetzel, since progressed to a “prevention-oriented” period, which is associated with current best practices and reflects the highest maturity level of the TMMi.

Furthermore, various industrial best-practices, practical experience using the TMM and testing surveys have contributed to the TMMi development providing it with its necessary empirical foundation and required level of practicality. They illustrate the current best and worst testing practices in the IT industry, and have allowed the developers of the TMMi framework to extract realistic benchmarks by which to evaluate and improve testing practices.

## 1.4 Scope of the TMMi

### 1.4.1 Software and System Engineering

The TMMi is intended to support testing activities and test process improvement in both the systems engineering and software engineering discipline. Systems engineering covers the development of total systems, which may or may not include software. Software engineering covers the development of software systems.

### 1.4.2 Test levels

Whereas some models for test process improvement focus mainly on high-level testing, e.g. TPI [Koomen and Pol] or address only one aspect of structured testing e.g. the test organization, The TMMi addresses all test levels (including static testing) and aspects of structured testing. With respect to dynamic testing, both low-level testing and high-level testing are within the scope of the TMMi. Studying the model more in detail one will learn that the model addresses all four cornerstones for structured testing (lifecycle, techniques, infrastructure and organization) [TMap].

### 1.4.3 TMMi and CMMI

It is also important to note that TMMi is positioned as a complementary model to the CMMI. In many cases a given TMMi level needs specific support from process areas at its corresponding CMMI level or from lower CMMI levels. In exceptional cases there is even a relationship to higher CMMI levels. Process areas and practices that are elaborated within the CMMI are mostly not repeated within TMMi; they are only referenced. For example the process area configuration management, which is of course also applicable to test (work) products / test ware, is not elaborated upon in detail within the TMMi; the practices from CMMI are referenced and implicitly re-used.

### 1.4.4 Assessments

Many organizations find value in benchmarking their progress in test process improvement for both internal purposes and for external customers and suppliers. Test process assessments focus on identifying improvement opportunities and understanding the organization’s position relative to the selected model or standard. The TMMi provides an excellent reference model to be used during such assessments. Assessment teams use TMMi to guide their identification and prioritization of findings. These findings with guidance of TMMi practices are used to plan improvements for the organization. The assessment framework itself is not part of the TMMi. Requirements for TMMi assessments are described by the TMMi Foundation in a separate document to be found at [www.TMMiFoundation.org](http://www.TMMiFoundation.org). These requirements are based upon the ISO 15504 standard. The achievement of a specific maturity level must mean the same thing for different assessed organizations. Rules for ensuring this consistency are contained in the TMMi assessment method requirements. The TMMi assessment method requirements contain guidelines for various classes of assessments, e.g. formal assessments, quick-scans and self-assessments.

### 1.4.5 Improvement Approach

The TMMi provide a full framework to be used as a reference model during test process improvement. It does not provide an approach for test process improvement such as the IDEAL<sup>4</sup> (Initiating, Diagnosing, Establishing, Acting, And Learning) model. Practical experiences have shown that the most powerful initial step to test process improvement is to build strong organizational sponsorship before investing in test process assessments. Given

<sup>4</sup> IDEAL is a registered service mark of Carnegie Mellon University



sufficient senior management sponsorship, establishing a specific, technically competent test process group that represents relevant stakeholders to guide test process improvement efforts had proved to be an effective approach. More information about the IDEAL model can be found at [www.sei.cmu.edu/ideal/ideal.html](http://www.sei.cmu.edu/ideal/ideal.html)

## 2 TMMi Maturity Levels

### 2.1 Overview

TMMi has a staged architecture for process improvement. It contains stages or levels through which an organization passes as its testing process evolves from one that is ad-hoc and unmanaged, to one that is managed, defined, measured, and optimized. Achieving each stage ensures that an adequate improvement has been laid as a foundation for the next stage. The internal structure of the TMMi is rich in testing practices that can be learned and applied in a systematic way to support a quality testing process that improves in incremental steps. There are five levels in the TMMi that prescribe a maturity hierarchy and an evolutionary path to test process improvement. Each level has a set of process areas that an organization needs to focus on to achieve maturity at that level. Experience has shown that organizations do their best when they focus their test improvement process efforts on a manageable number of process areas at a time, and that those areas require increasing sophistication as the organization improves. Because each maturity level forms a necessary foundation for the next level, trying to skip a maturity level is usually counter productive. At the same time, you must recognize that test process improvement efforts should focus on the needs of the organization in the context of its business environment and the process areas at higher maturity levels may address the current needs of an organization or project. For example, organizations seeking to move from maturity level 1 to maturity level 2 are frequently encouraged to establish a test group, which is addressed by the Test Organization process area that resides at maturity level 3. Although the test group is not a necessary characteristic of a TMMi level 2 organization, it can be a useful part of the organization's approach to achieve TMMi maturity level 2.

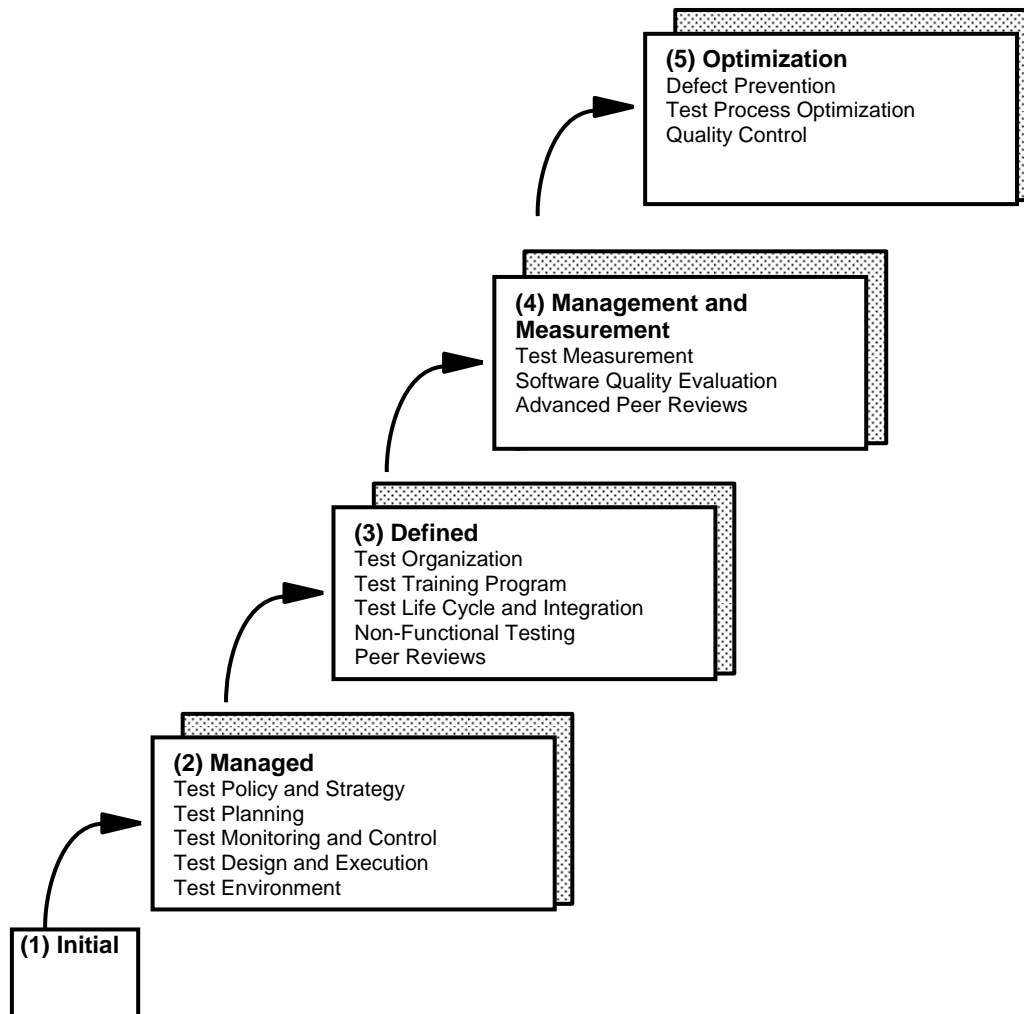


Figure 1: TMMi maturity levels and process areas

The process areas for each maturity level of the TMMi are shown in figure 1. They are fully described later in other chapters and are also listed below along with a brief description of the characteristics of an organization at each TMMi level. The description will introduce the reader to the evolutionary path prescribed in the TMMi for test process improvement.

Note that the TMMi does not have a specific process area dedicated to test tools and/or test automation. Within TMMi test tools are treated as a supporting resource (practices) and are therefore part of the process area where they provide support, e.g. applying a test design tool is a supporting test practice within the process area Test Design and Execution at TMMi level 2 and applying a performance testing is tool is a supporting test practice within the process area Non-Functional Testing at TMMi level 3.

## 2.2 Level 1 Initial

At TMMi level 1, testing is a chaotic, undefined process and is often considered a part of debugging. The organization usually does not provide a stable environment to support the processes. Success in these organizations depends on the competence and heroics of the people in the organization and not the use of proven processes. Tests are developed in an ad-hoc way after coding is completed. Testing and debugging are interleaved to get the bugs out of the system. The objective of testing at this level is to show that the software runs without major failures. Products are released without adequate visibility regarding quality and risks. In the field, the product does often not fulfill its needs, is not stable, or is too slow to work with. Within testing there is a lack of resources, tools and well-educated staff. At TMMi level 1 there are no defined process areas. Maturity level 1 organizations are characterized by a tendency to over commit, abandonment of processes in a time of crises, and an inability to repeat their successes. Also products tend not to be released on time, budgets are overrun and quality is not according to expectations.

## 2.3 Level 2 Managed

At TMMi level 2, testing becomes a managed process and is clearly separated from debugging. The process discipline reflected by maturity level 2 helps to ensure that existing practices are retained during times of stress. However, testing is by many stakeholders still perceived as being a project phase that follows coding. In the context of improving the test process, a company-wide or program-wide test strategy is established. Test plans are also being developed. Within the test plan a test approach is defined, whereby the approach is based on the result of a product risk assessment. Risk management techniques are used to identify the product risks based on documented requirements. The test plan defines what testing is required, when, how and by whom. Commitments are established with stakeholders and revised as needed. Testing is monitored and controlled to ensure it is going according to plan and actions can be taken if deviations occur. The status of the work products and the delivery of testing services are visible to management. For deriving and selecting test cases from specifications test design techniques are applied. However, testing may still start relatively late in the development lifecycle, e.g. during the design or even during the coding phase. Testing is multi-leveled: there are unit, integration, system and acceptance test levels. For each identified test level there are specific testing objectives defined in the organization-wide or program-wide test strategy. The main objective of testing in a TMMi level 2 organizations is to verify that the product satisfies the specified requirements. The purpose is also to clearly differentiate the processes of testing and debugging. Many quality problems at this TMMi level occur because testing occurs late in the development lifecycle. Defects are propagated from the requirements and design into code. There are no formal review programs as yet to address this important issue. Post code, execution based testing is by many stakeholders still considered the primary testing activity.

The process areas at TMMi level 2 are:

- 2.1 Test Policy and Strategy
- 2.2 Test Planning
- 2.3 Test Monitoring and Control
- 2.4 Test Design and Execution
- 2.5 Test Environment

## 2.4 Level 3 Defined

At TMMi level 3, testing is no longer a phase that follows coding. It is fully integrated into the development lifecycle and the associated milestones. Test planning is done at an early project stage, e.g. during the requirements phase, by means of a master test plan. The development of a master test plan builds on the test planning skills and commitments acquired at TMMi level 2. The organization's set of standard test processes, which is the basis for maturity level 3, is established and improved over time. A test organization and a specific test training program exist, and testing is perceived as being a profession. Test process improvement is fully institutionalized as part of the test organization. Organizations at this level understand the importance of reviews in quality control; a formal review program is implemented although not yet fully linked to the dynamic testing process. Reviews take place across the lifecycle. Test professionals are involved in reviews on requirements specifications. Whereby the test designs at TMMi level 2 focus mainly on functionality testing, test designs and test techniques are expanded, depending the business objectives, to also include non-functional testing, e.g. on usability and/or reliability.

A critical distinction between TMMi maturity level 2 and 3 is the scope of the standards, process descriptions, and procedures. At maturity level 2 these may be quite different in each specific instance, e.g. on a particular project. At maturity level 3 these are tailored from the organization's set of standard processes to suit a particular project or organizational unit and therefore are more consistent except for the differences allowed by the tailoring guidelines. Another critical distinction is that at maturity level 3, processes are typically described more rigorously than at maturity level 2. As a consequence at maturity level 3, the organization must revisit the maturity level 2 process areas.

The process areas at TMMi level 3 are:

- 3.1 Test Organization
- 3.2 Test Training Program
- 3.3 Test Lifecycle and Integration
- 3.4 Non-Functional Testing
- 3.5 Peer Reviews

## 2.5 Level 4 Management and Measurement

In TMMi 4 organizations testing is a thoroughly defined, well-founded and measurable process. At maturity level 4, the organization and projects establish quantitative objectives for product quality and process performance and use them as criteria in managing them. Product quality and process performance is understood in statistical terms and is managed throughout the lifecycle. Measures are incorporated into the organization's measurement repository to support fact-based decision making. Reviews and inspections are considered to be part of testing and used the measure document quality. The static and dynamic test approaches are integrated into one. Reviews are formally used as means to control quality gates. Products are evaluated using quantitative criteria for quality attributes such as reliability, usability and maintainability. An organization wide test measurement program provides information and visibility regarding the test process. Testing is perceived as evaluation; it consists of all lifecycle activities concerned with checking products and related work products.

The process areas at TMMi level 4 are:

- 4.1 Test Measurement
- 4.2 Product Quality Evaluation
- 4.3 Advanced Peer Reviews

## 2.6 Level 5 Optimization

On the basis of all results that have been achieved by fulfilling all the improvement goals of the previous maturity levels, testing is now a completely defined process and one is capable of controlling the costs and the testing effectiveness. At TMMi maturity level 5, an organization continually improves its processes based on a quantitative understanding of the common cause of variation inherent in processes. Improving test process performance is carried out through incremental and innovative process and technological improvements. The methods and techniques are optimized and there is a continuous focus on fine-tuning and test process improvement. Defect prevention and quality control are practiced. Statistical sampling, measurements of confidence levels, trustworthiness, and reliability drive the test process. Amongst others "Defect Prevention" and "Quality Control" are introduced as process areas. The test process is characterized by sampling based quality measurements. A detailed procedure exists for selecting and evaluating test tools. Tools support the test process as much as possible during test design, test execution, regression testing, test case management, etc. Process re-use is also practiced at level 5 supported by a process asset library. Testing is a process with the objective to prevent defects.

Process areas at level 5 are:

- 5.1 Defect Prevention
- 5.2 Test Process Optimization
- 5.3 Quality Control

## 3 Structure of the TMMi

The structure of the TMMi is largely based on the structure of the CMMI. This is a major benefit because many people/organizations are already familiar with the CMMI structure. The CMMI structure makes a clear distinction between practices that are required (goals) or recommended (specific practices, typical work products, etc.) to implement. This aspect is also included in the TMMi. In this chapter, the components and structure of the TMMi structure are described. In addition the support provided by the CMMI to a TMMi implementation is described.

### 3.1 Required, Expected and Informative Components

The various components are grouped into three categories: required, expected and informative.

#### 3.1.1 Required Components

Required components describe what an organization must achieve to satisfy a process area. This achievement must be visibly implemented in an organization's processes. The required components in TMMi are the specific and generic goals. Goals satisfaction is used in assessments as the basis for deciding if a process area has been achieved and satisfied.

#### 3.1.2 Expected Components

Expected components describe what an organization will typically implement to achieve a required component. Expected components guide those who implement improvements or perform assessments. Expected components include both specific and generic practices. Either the practices as described or acceptable alternatives to the practices must be present in the planned and implemented processes of the organization, before goals can be considered satisfied.

#### 3.1.3 Informative Components

Informative components provide details that help organizations get started in thinking about how to approach the required and expected components. Sub-practices, typical work products, notes, examples, and references are all informative model components.

## 3.2 Components of the TMMi

The TMMi model required and expected components can be summarized to illustrate their relationship as in figure 2. The following sections provide a description of the components. Note that the TMMi also provides a specific glossary of terms. The terms used in the glossary are largely re-used from the international test terminology standard developed by the International Software Testing Qualifications Board (ISTQB): Standard glossary of terms used in Software Testing [ISTQB].

### 3.2.1 Maturity levels

A maturity level within the TMMi can be regarded as a degree of organizational test process quality. It is defined as an evolutionary plateau of test process improvement. Each level progressively develops an important part of the organization's test processes. There are five maturity levels within the TMMi. Each maturity level tells what to implement in order to achieve the given level. The higher the maturity level the organization is ranked at, the more mature the test process of the organization is. To reach a particular maturity level, an organization must satisfy all of the appropriate goals (both specific and generic) of the process areas at the specific level and also those at earlier maturity levels. Note that all organizations possess a minimum of TMMi level 1, as this level does not contain any goals that must be satisfied.

### 3.2.2 Process Areas

As stated with the exception of level 1, each maturity level consists of several process areas that indicate where an organization should focus to improve its test process. Process areas identify the issues that must be addressed to achieve a maturity level. Each process area identifies a cluster of test related activities. When the practices are all performed a significant improvement in activities related to that area will be made. In the TMMi, only those process areas are identified that are considered to be key determinants of test process capability. All process areas of the maturity level and the lower maturity levels must be satisfied to consider a maturity level to be achieved. For example, if an organization is at TMMi level 3, it has satisfied all of the process areas at both TMMi level 2 and TMMi level 3.

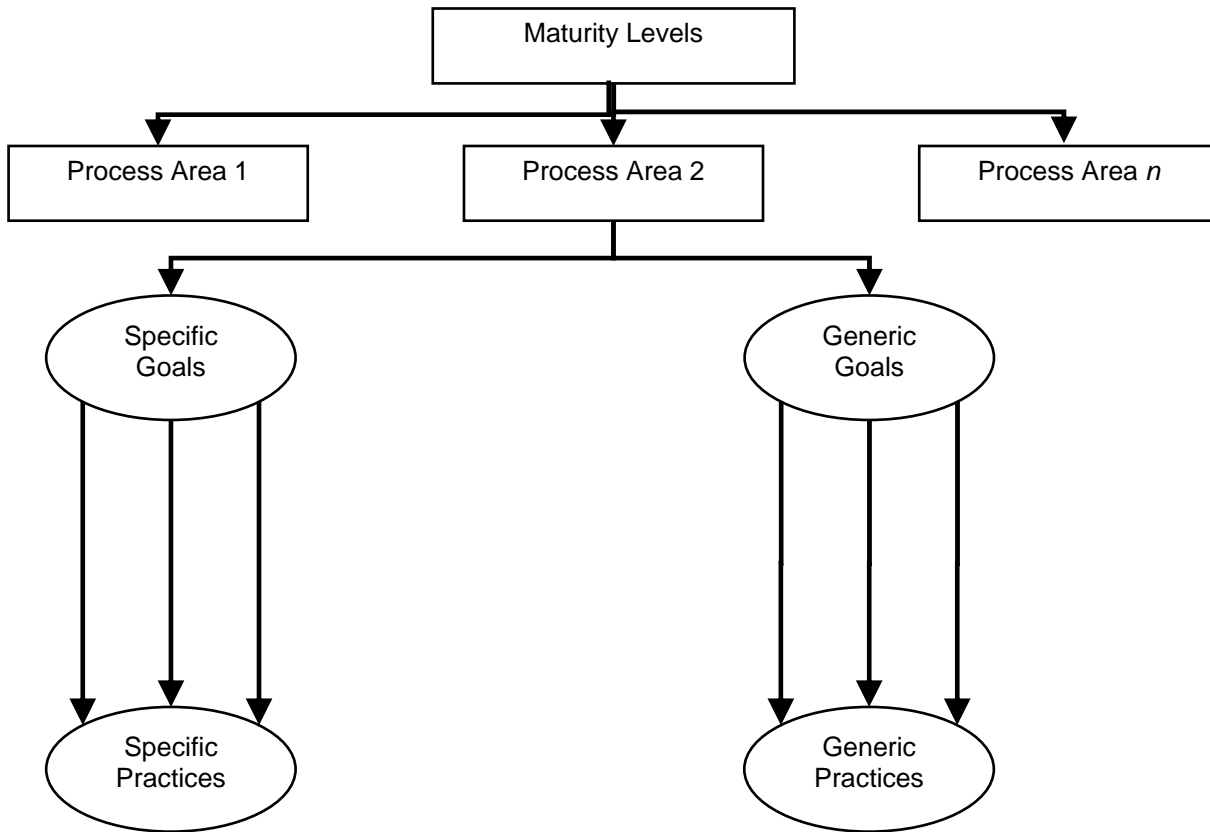


Figure 2: TMMi structure and components

### 3.2.3 Purpose

The purpose statement describes the purpose of the process area and is an informative component. For example, the purpose statement of the test planning process area is “define a test approach based on the identified risks and the defined test strategy, and to establish and maintain well-founded plans for performing and managing the testing activities”.

### 3.2.4 Introductory Notes

The introductory notes section of the process area describes the major concepts covered in the process area and is an informative component.

### 3.2.5 Scope

The scope section of the process area specifically identifies the test practices that are addressed by the process area, and if necessary test practices that are explicitly outside the scope of this process area.

### 3.2.6 Specific Goals

A specific goal describes the unique characteristic that must be present to satisfy the process area. A specific goal is a required model component and is used in assessments to help determine whether a process area is satisfied.

### 3.2.7 Generic Goals

Generic goals appear near the end of a process area and are called ‘generic’ because the same goal statement appears in multiple process areas. A generic goal describes the characteristics that must be present to institutionalize the processes that implement a process area. A generic goal is a required model component and is used in assessments to help determine whether a process area is satisfied.

### 3.2.8 Specific practices

A specific practice is the description of an activity that is considered important in achieving the associated specific goal. The specific practice describes the activities expected to result in achievement of the specific goals of a process area. A specific practice is an expected model component.

### 3.2.9 Typical Work Products

The typical work products section lists sample outputs from a specific practice. These examples are called 'typical work products' because there are often work products that are just as effective but are not listed. A typical work product is an informative model component.

### 3.2.10 Sub-practices

A sub-practice is a detailed description that provides guidance for interpreting and implementing a specific practice. Sub-practices may be worded as if prescriptive, but are actually an informative component meant only to provide ideas that may be useful for test process improvement.

### 3.2.11 Generic Practices

Generic practices appear near the end of a process area and called 'generic' because the same practice appears in multiple process areas. A generic practice is the description of an activity that is considered important in achieving the associated generic goal. A generic practice is an expected model component.

### 3.2.12 Generic Practices Elaborations

Generic practices elaboration appears after a generic practice in a process area to provide guidance on how the generic practice should be applied uniquely to the process area. A generic practice elaboration is an informative model component.

### 3.2.13 Supporting Informative Components

There are many places where further information is needed to describe a concept. This informative information is provided in terms of the following components:

#### 3.2.13.1 Notes

A note is a text that can accompany any other model component. It may provide detail, background, or rationale. A note is an informative model component.

#### 3.2.13.2 Examples

An example is a component comprising text and often a list of items, usually in a box, that can accompany nearly any other component and provides one or more examples to clarify a concept or described activity. An example is an informative model component.

#### 3.2.13.3 References

A reference is a pointer to additional or more detailed information in related process areas and can accompany nearly any other model component. A reference is an informative model component.

## 3.3 Generic Goals and Generic Practices

This section describes all of the generic goals and generic practices. The generic goals and generic practices are largely derived from the CMMI. The generic goals are organized in numeric order. The generic practices are also organized in numerical order under the generic goal they support. Note that the generic goal from the CMMI, GG1 'Achieve Specific Goals' is not taken into account since this only relates to the continuous representation of the CMMI and therefore has no relevance to the staged representation of the TMMi. Otherwise the numbering scheme of the CMMI is fully adopted to avoid confusion for organizations using both CMMI and TMMi.

The capability level you are targeting at will determine which generic goals and practices are applicable. When trying to reach maturity level 2 the process areas at maturity level 2 as well as generic goal 2 and the accompanying generic practices are applicable. Generic goal 3 is only applicable when trying to reach maturity level 3 or higher. This means that when you have already achieved a maturity level 2 rating, to achieve a maturity level 3 rating you must return to maturity level 2 process areas and apply generic goal 3 and the accompanying practices.

Institutionalization is an important concept in process improvement. When mentioned in the generic goal and generic practice descriptions, institutionalization implies that the process is ingrained in the way the work is performed and there is commitment and consistency to performing the process. An institutionalized process is more likely to be retained during times of stress. When the requirements and objectives for the process change, however, the implementation of the process may also need to change to ensure that it remains active. The generic practices describe activities that address these aspects of institutionalization.

## GG 2 Institutionalize a Managed Process

A managed process is a process that accomplishes the work necessary to produce work products; is planned and executed in accordance with policy; employee skilled people have adequate resources to



produce controlled outputs; involves relevant stakeholders; is monitored and controlled, reviewed; and is evaluated for adherence to its process descriptions. The process may be instantiated by a project, group, or organizational unit. The control provided by a managed process helps to ensure that the established process is retained during times of stress.

### **GP 2.1 Establish an organizational policy**

The purpose of this generic practice is to define the organizational expectations for the process and make these expectations visible to those in the organization who are affected. In general, senior management is responsible for establishing and communicating guiding principles, direction, and expectations for the organization.

### **GP 2.2 Plan the process**

The purpose of this generic practice is to determine what is needed to perform the process and to achieve the established objectives, to prepare a plan for performing the process, to prepare a process description, and to get agreement on the plan from relevant stakeholders by performing reviews.

### **GP 2.3 Provide resources**

The purpose of this generic practice is to ensure that resources necessary to perform the process as defined by the plan are available when they are needed. Resources include adequate funding, appropriate physical facilities, skilled people, and appropriate tools.

### **GP 2.4 Assign responsibilities**

The purpose of this generic practice is to ensure that there is accountability for performing the process and achieving the specified results throughout the life of the process. The people assigned must have the appropriate authority to perform the assigned responsibilities. Responsibilities can be assigned using detailed job descriptions or in living documents, such as the plan for performing the process.

### **GP 2.5 Train people**

The purpose of this generic practice is to ensure that the people have the necessary skills and expertise to perform or support the process. Appropriate training is provided to the people who will perform the work. Overview training is provided to orient people who interact with those performing the work. Training supports the successful performance of the process by establishing a common understanding of the process, and by imparting the skills and knowledge needed to perform the process.

### **GP 2.6 Manage configuration**

The purpose of this generic practice is to establish and maintain the integrity of the designated work products of the process throughout their useful life. The designated work products are specifically identified in the plan for performing the process, along with a specification of the level of configuration management, e.g. version control or formal configuration management using baselines. Examples of configuration management practices include version control, change history and control, status identification and usage of configuration management tools for storage. Refer to the Configuration Management process area within CMMI for more information on placing work products under configuration management.

### **GP 2.7 Identify and involve relevant stakeholders**

The purpose of this generic practice is to establish and maintain the expected involvement of stakeholders during the execution of the process. Relevant stakeholders are involved in activities such as planning, decisions, commitments, communications, reviews and resolution of problems. Critical stakeholders in the testing process include managers and users/customer. The manager's role involves commitment and ability to perform activities and tasks related to improving testing capability. The user's or customer's role involves co-operation, support or even performing testing activities. Users/customers should be involved in quality-related activities and tasks that concern user-oriented needs. The focus is on solicitation user/customer support, consensus and participating in activities such as product risk analysis, acceptance testing and possibly usability testing. Depending on the test level the developer may also be a stakeholder, e.g. at component testing the developer often performs the testing activities himself, however at acceptance test level the developer becomes a stakeholder for discussing incidents found, agreeing on entry criteria etc.



**GP 2.8 Monitor and control the process**

The purpose of this generic practice is to perform the direct day-to-day monitoring and controlling of the test process. Appropriate visibility into the test process is maintained so that appropriate corrective action can be taken when necessary. Monitoring and controlling the process involves measuring appropriate attributes of the test process and work products produced by the test process. Refer to the Test monitoring and control process area for more information on placing work products under configuration management. Refer to the Measurement and Analysis process area with CMMI for more information on measurement.

**GP 2.9 Objectively evaluate adherence**

The purpose of this generic practice is to provide credible assurance that the process is implemented as planned and adheres to its process description, standard, and procedures. People not directly responsible for managing or performing the activities of the test process typically evaluate adherence. In many cases, adherence is evaluated by people within the organization, but external to the test process or project. Refer to the Process and Product Quality Assurance process area with CMMI for more information on objectively evaluating adherence.

**GP 2.10 Review status with higher level management**

The purpose of this generic practice is to provide higher level management with the appropriate visibility into the process. Higher level management includes those levels of management in the organization above the immediate level of management responsible for the process. These reviews are for managers who provide policy and overall guidance for the process, not for those who perform the direct day-to-day monitoring and controlling of the process.

**GG 3 Institutionalize a Defined Process**

A defined process is a managed process that is tailored from the organization’s set of standard processes according to the organization’s tailoring guidelines; has maintained process descriptions; and contributes work products, measures, and other process improvement information to the organizational process assets. A critical distinction between a managed process and a defined process is the scope of application of the process descriptions, standards, and procedures. For a managed process descriptions, standards, and procedures are applicable to a particular project, group, or organizational function. As a result, the managed processes of two projects in one organization may be different. A defined process is standardized as much as possible across the organization and only when required adapted for a specific project or organizational function based on tailoring guidelines.

**GP 3.1 Establish a defined process**

The purpose of this generic practice is to establish and maintain a description of the process that is tailored from the organization’s set of standard processes to address the needs of a specific instantiation. The organization should have standard processes that cover the process area, as well as have guidelines for tailoring these standard processes to meet the needs of a project or organizational function. With a defined process, variability in how the processes are performed across the organization is reduced and process assets, data, and learning can be effectively shared. Refer to the Organization Process Definition process area with CMMI for more information about the organization’s set of standard processes and tailoring guidelines.

**GP 3.2 Collect improvement information**

The purpose of this generic practice is to collect information and artifacts derived from planning and performing the process to support future use and improvement of the organization’s processes and process assets. The information and artifacts are stored and made available to those who are (or who will be) planning and performing the same or similar processes.

**3.4 Supporting process areas for generic practices**

While generic goals and generic practices are the model components that directly address the institutionalization of a process across the organization, many process areas either in TMMi or CMMI likewise address institutionalization by supporting the implementation of the generic practices. The table hereafter provides an overview of the process areas that partly or fully support the implementation of a generic practice.

Generic Practice	Supporting Process Area
------------------	-------------------------

Generic Practice	Supporting Process Area
GP2.2 Plan the process	Test Planning; the Test Planning process can implement GP2.2 in full for all project related process areas (except for test planning itself). Test planning itself can be addressed as part of the CMMI process area Project Planning.
GP2.5 Train People	Test Training Program; the Test Training Program process supports the implementation of GP2.5 for all process areas by making the organization-wide training program available to those who will perform or support the processes. In addition the Test Planning process may support this generic practice by identifying and organizing the training needs that are needed for testing in the project and documenting those in the test plan.
G2.6 Manage configuration	Configuration Management; the CMMI Configuration Management process can implement GP2.6 in full for all project-related process areas as well as some of the organizational process areas.
G2.7 Identify and involve the relevant stakeholders	Test Planning; the Test Planning process may support this generic practice for all project related process areas by planning the involvement of identified stakeholders and documenting those in the test plan. Stakeholder involvement for test planning itself can be addressed as part of the CMMI process area Project Planning.
GP2.8 Monitor and control the Process	Test Monitoring and Control; the Test Monitoring and Control process area can implement GP2.8 in full for all process areas.
GP2.9 Objectively evaluate adherence	Process and Product Quality Assurance; the CMMI Process and Product Quality Assurance process can implement GP2.9 in full for all process areas.

Table 1: Supporting process areas for generic practices

### 3.5 Supporting CMMI process areas for TMMi

Although TMMi can be used in isolated, it is also positioned as a complementary model to the CMMI. As a result in many cases a given TMMi level needs specific support from process areas at its corresponding CMMI level or from higher CMMI levels. Process areas and practices that are elaborated within the CMMI are mostly not repeated within TMMi; they are only referenced. An overview of supporting CMMI process areas required for TMMi level 2 achievement is shown in table 2. Supporting CMMI process areas required for TMMi level 3 achievement are shown in table 3. Note that some these relationship were already identified, although from a different perspective, in the previous section.

TMMi	CMMI	Supporting CMMI process areas
2	2	<p><i>Configuration Management</i>, as stated above the Configuration Management process area can implement GP2.6 in full for all project-related process areas as well as some of the organizational process areas.</p> <p><i>Process and Product Quality Assurance</i>, as stated above Process and Product Quality Assurance process can implement GP2.9 in full for all process areas.</p> <p><i>Project Control and Monitor</i>, this process area will provide support for the implementation of the TMMi process area “Test Monitoring and Control”. Project management practices can be re-used for test management.</p> <p><i>Project Planning</i>, this process area will provide support for the implementation of the TMMi process area “Test Planning”. Project management practices can be re-used for test management. Project planning will specifically support the implementation of the generic practice GP2.7 regarding stakeholder involvement for Test Planning.</p> <p><i>Measurement and Analysis</i>, this process area will provide support for the implementation of the SG 3 ‘Establish test performance indicators’ of the TMMi process area “Test Policy and Strategy”.</p> <p><i>Requirements Management</i>, the implementation this process area is a constraint for managing derived (work) products, such as the product risk analysis and test designs, and keeping them up-to-date. The practices regarding maintaining traceability can possibly be re-used within the “Test Design and Execution” process area.</p>

TMMi	CMMI	Supporting CMMI process areas
2	3	<p><i>Requirements Development</i>, practices from this process area can be re-used when developing test environment requirements within the process area “Test Environment”.</p> <p><i>Risk Management</i>, practices from this process area can be re-used for identifying and controlling product risk and test project risks within the process areas “Test Planning” and “Test Monitoring and Control”.</p>

Table 2: Support for TMMi maturity levels from CMMI process areas

TMMi	CMMI	Supporting CMMI process areas
3	2	<p><i>Configuration Management</i>, the Configuration Management process area can implement GP2.6 in full for all project-related process areas as well as some of the organizational process areas.</p> <p><i>Process and Product Quality Assurance</i>, the Process and Product Quality Assurance process can implement GP2.9 in full for all process areas.</p> <p><i>Project Planning</i>, this process area will provide support for the implementation of the TMMi process area “Test Lifecycle and Integration”, especially SG3 ‘Establish a master test plan’. Project management practices can be re-used for test management.</p>
3	3	<p><i>Organization Process Focus</i>, this process area will provide support for the implementation of the TMMi process area “Test Organization”, especially for SG4 ‘Determine, plan and implement test process improvements’ and SG5 ‘Deploy organizational test processes and incorporate lessons learned’.</p> <p><i>Organization Process Definition</i>, this process area will provide support for the implementation of the TMMi process area “Test Lifecycle and Integration”, especially for SG1 “Establish organizational test process assets”.</p> <p><i>Organization Training</i>, this process area will provide support for the implementation of the TMMi process area “Test Training Program”.</p> <p><i>Verification</i>, the practices with SG2 ‘Perform peer reviews’ of this process area will provide support for the implementation of the TMMi process area “Peer Reviews”.</p>

Table 3: Support for TMMi maturity levels from CMMI process areas

Note that the test specific process areas of the CMMI Verification and Validation are not listed as supporting process areas for the dynamic testing processes within TMMi. For these CMMI process areas, the TMMi process areas provide support and a more detailed specification of what is required to establish a defined verification and validation process.

## TMMi Level 2: Managed

In level 2, testing becomes a managed process and is clearly separated from debugging. The process discipline reflected by maturity level 2 helps to ensure that existing practices are retained during times of stress. However, it is by many stakeholders still perceived as being a project phase that follows coding. In the context of improving the test process, a company-wide or program-wide test strategy is established. Test plan are also being developed. Within the test plan a test approach is defined, whereby the approach is based on the level of risk. Risk management techniques are used to identify the product risks based on documented requirements. The test plan will define what testing is required, when, how and by whom. Commitments are established among relevant stakeholders and revised as needed. Testing is monitored and controlled to ensure it is going according to plan and actions can be taken if deviations occur. The status of the work products and the delivery of testing services are visible to management. For deriving and selecting test cases from specifications test design techniques are applied. However, testing may still start relatively late in the development lifecycle, e.g. during the design or even during the coding phase. Testing is multi-leveled: there are unit, integration, system and acceptance test levels. For each identified test level there are specific testing objectives defined in the organization-wide or program-wide test strategy. The main objective of testing in a TMMi level 2 organizations is to verify that the product satisfies the specified requirements. The purpose is also to clearly differentiate the processes of testing and debugging. Many quality problems at this TMMi level occur because testing occurs late in the development lifecycle. Defects are propagated from the requirements and design into code. There are no review programs as yet to address this important issue. Post code, execution based testing is by many stakeholders still considered the primary testing activity.

The process areas at TMMi level 2 are:

- 2.1 Test Policy and Strategy
- 2.2 Test Planning
- 2.3 Test Monitoring and Control
- 2.4 Test Design and Execution
- 2.5 Test Environment

## PA2.1 Test Policy and Strategy

### Purpose

The purpose of Test Policy and Strategy is to develop and establish a test policy and an organization-wide or program-wide test strategy in which the test levels are unambiguously defined. To measure test performance, test performance indicators are introduced.

### Introductory Notes

When an organization wants to improve its test process, it should first clearly define a test policy. The test policy defines the organization's overall test objectives, goals and strategic views regarding testing. It is important that the test policy is aligned with the overall business (quality) policy of the organization. A test policy is necessary to attain a common view on testing between all stakeholders within an organization. This common view is indispensable to align test (process improvement) activities. The test policy should address both new development and maintenance testing activities. Within the test policy the objectives for test process improvement should be stated. These objectives will subsequently be translated into a set of key test performance indicators. The test policy and the accompanying performance indicators provide a clear direction, and a means to communicate expected and achieved levels of test performance. The performance indicators have the objective to show the value of testing and test process improvement to stakeholders.

Based upon the test policy a test strategy will be defined. The test strategy covers the generic test requirements for an organization or program (one or more projects). The test strategy addresses the generic product risks and presents a process for mitigating those risks in line with the testing policy. The test strategy therefore starts by performing a generic product risk assessment studying the products being developed within a program or organization. A typical test strategy will include a description of the test levels that are to be applied, for example: unit, integration, system and acceptance test. For each test level amongst others the objectives, responsibilities, main tasks and entry/exit criteria are defined. The test strategy serves as a starting point for the testing activities within projects. The projects are set up in accordance with the organization-wide or program-wide test strategy. When a test strategy is defined and followed, less overlap between the test levels is likely to occur leading to a more efficient test process. Also since the test objectives and approach of the various levels is aligned fewer holes are likely to remain leading to a more effective test process.

### Scope

The process area Test Policy and Strategy involves the definition and deployment of a test policy and test strategy. Within the test strategy, test levels are identified. For each test level, amongst other test objectives, responsibilities and main tasks are defined. To measure test performance and the accomplishment of test (improvement) goals, test performance indicators are defined and deployed.

### Specific Goal and Practice Summary

- SG1 Establish a test policy
  - SP 1.1 Define test goals
  - SP 1.2 Define test policy
  - SP 1.3 Distribute the test policy to stakeholders
- SG2 Establish a test strategy
  - SP 2.1 Perform a generic product risk assessment
  - SP 2.2 Define test strategy
  - SP 2.3 Distribute the test strategy to stakeholders
- SG3 Establish test performance indicators
  - SP 3.1 Define test performance indicators
  - SP 3.2 Deploy test performance indicators

## Specific Practices by Goals

### SG 1 Establish a Test Policy

*A test policy, aligned with the business (quality) policy, is established and agreed by the stakeholders.*

#### SP 1.1 Define test goals

*Define and maintain test goals based upon business needs and objectives.*

##### **Typical work products**

1. Test goals

##### **Sub-practices**

2. Study business needs and objectives

*Examples of business needs and objectives to be studied include the following:*

- Mission statement
- Business and user needs regarding the products
- Business drivers
- Main goals of a quality program
- Business (quality) policy
- Type of business, e.g. risk level of products being developed

3. Provide feedback for clarifying business needs and objectives as necessary
4. Define test goals traceable to business needs and objectives

*Examples of test goals include the following:*

- Validate products for 'fit-for-use'
- Prevent defects for occurring in operation
- Verify compliance to external standards
- Provide visibility regarding product quality
- Short test execution lead-time

5. Review the test goals with stakeholders
6. Revisit and revise the test goals as appropriate, e.g. on a yearly basis

#### SP 1.2 Define the test policy

*A test policy, aligned with the business (quality) policy, is defined and agreed by the stakeholders.*

##### **Typical work products**

1. Test policy

##### **Sub-practices**

1. Define the test policy based on the defined test goals

*Examples of statements to be part of a test policy typically include the following:*

- A definition of testing
- A definition of debugging (fault localization and repair)
- Basic view points regarding testing and the testing profession
- The objectives and added value of testing
- Quality levels to be achieved
- The level of independence of the test organization
- High level test process definition
- The key responsibilities of testing
- The organizational approach to and objectives of test process improvement

2. Clearly separate testing from debugging within the test policy
3. Review the test policy with stakeholders
4. Define and establish ownership for test policy



5. Revisit and revise the test policy as appropriate, e.g. on a yearly basis

### SP 1.3 Distribute the test policy to stakeholders

*The test policy and test goals are presented and explained to stakeholders inside and outside testing.*

#### **Typical work products**

1. Deployment plan
2. Presentation test policy

*Examples of distribution mechanisms include the following:*

- Documenting it in a handbook (quality system)
- Presentations in project and/or departmental meetings
- Posters on the wall
- Making it part of the departmental introduction program
- Access on a central web portal

## SG 2 Establish a Test Strategy

*An organization-wide or program-wide test strategy is established and deployed; identifying and defining the test levels to be performed.*

### SP 2.1 Perform a generic product risk assessment

*A generic product risk assessment is performed to identify the typical critical areas for testing.*

#### **Typical work products**

1. Generic product risk list, with a category and priority assigned to each risk

#### **Sub-practices**

1. Identify and select stakeholders that need to contribute to the generic risk assessment
2. Identify generic product risks using input from stakeholders
3. Document the context and potential consequences of the generic product risk
4. Identify the relevant stakeholders associated for each generic product risk
5. Analyze the identified generic products risks using the predefined parameters, e.g. likelihood and impact
6. Categorize and group generic product risks according to the defined risk categories
7. Prioritize the generic product risks for mitigation
8. Review and obtain agreement with stakeholders on the completeness, category and priority level of the generic product risks
9. Revise the generic product risks as appropriate

Note that product risk categories and parameters as defined in the “Test Planning” process area (SP 1.1 “Define product risk categories and parameters”) are largely re-used within this specific practice.

Refer to SG 1 “Perform a product risk assessment” from the process area “Test Planning” for more details on the (sub) practices for performing the generic product risk assessment.

### SP 2.2 Define test strategy

*The test strategy is defined, identifying the test levels and defining for each level amongst others the objectives, responsibilities, main tasks and entry/exit criteria.*

#### **Typical work products**

1. Test strategy

#### **Sub-practices**

1. Study test policy and goals
2. Provide feedback for clarifying test policy and goals as necessary
3. Define the test strategy clearly linked with the defined test policy and goals

*Examples of topics to be addressed as part of a test strategy include the following:*

- Generics risks of the products being developed

- Overall test model (V-model, incremental lifecycle) as a way to mitigate the risks
- Test levels (e.g. unit, integration, system and acceptance test)
- Objectives, responsibilities and main tasks at each test level, e.g.
  - for unit testing:
    - verifying that the unit operates as specified in the unit design a certain level of code coverage is achieved
  - for integration testing:
    - verifying that the units together operate as specified in the global design
    - verifying that the interfaces operate as specified in the interface specification
  - for system testing:
    - verifying that the system operates as specified in the requirements specification
    - a certain level of system requirements coverage is achieved
  - for acceptance testing:
    - verifying that the system satisfies acceptance criteria
    - validating whether the system is 'fit for use'
    - a certain level of user requirements coverage is achieved
- Test case design techniques to be used at each test level
- Test types to be carried out at each test level
- Entry and exit criteria for each test level
- Standards that must be complied with
- Level of independence
- Environment in which the tests will be executed
- Approach to automation at each test level
- Approach to regression testing

4. Review the test strategy with stakeholders

5. Define and establish ownership for test strategy

6. Revisit and revise the test strategy as appropriate, e.g. on a yearly basis

Note that the test strategy will serve as a starting point for testing to be performed in a project. However, each project can tailor the overall strategy to its needs. Non-compliances shall be clearly documented in the test plan.

### **SP 2.3** Distribute the test strategy to the stakeholders

*The test strategy is presented to and discussed with stakeholders inside and outside testing.*

#### **Typical work products**

1. Deployment plan
2. Presentation test strategy

*Examples of distribution mechanisms include the following:*

- Documenting it in a handbook and/or quality system
- Presentations in project and/or departmental meetings
- Posters on the wall
- Making it part of the departmental introduction program
- Access on a central web portal

### **SG 3** Establish Test Performance Indicators

*A set of goal oriented test process performance indicators is established and deployed*

#### **SP 3.1** Define test performance indicators

*The test performance indicators are defined based upon the test policy and goals, including the procedure for data collection, storage and analysis.*

#### **Typical work products**

1. Test performance indicators



2. Data collection, storage, analysis and reporting procedure

### **Sub-practices**

1. Study test policy and goals, e.g. the objectives for test process improvement
2. Provide feedback for clarifying test policy and goals as necessary
3. Define the test performance indicators traceable to the test policy and goals

*Examples of test performance indicators include the following:*

- Test effort and cost
- Test lead time
- Number of defects found
- Defect detection percentage
- Test maturity level

4. Review the performance indicators with stakeholders
5. Define and establish ownership for test performance indicators
6. Specify how performance indicators will be obtained and stored
7. Specify how performance indicators will be analyzed and reported

## **SP 3.2 Deploy test performance indicators**

*Deploy the test performance indicators and provide measurement results, which address the identified test performance indicators, to stakeholders.*

### **Typical work products**

1. Test performance indicators data
2. Reports on test performance indicators

### **Sub-practices**

1. Obtain specified performance indicators data
2. Analyze and interpret performance indicators
3. Manage and store performance indicators data and analysis results
4. Report the performance indicators to stakeholders on a periodic basis
5. Assist stakeholders in understanding the results

*Examples of actions to assist in understanding of results include the following:*

- Discussing the results with relevant stakeholders
- Provide contextual information that provides background and explanation

## **Generic Practices by Goals**

### **GG 2 Institutionalize a Managed Process**

#### **GP 2.1 Establish an organizational policy**

*Establish and maintain an organizational policy for planning and performing the test policy and strategy process.*

#### **Elaboration**

Typically, at an organizational level it is documented that on a periodic basis, e.g. yearly, the test policy and test strategy will be revisited and if necessary updated.

#### **GP 2.2 Plan the process**

*Establish and maintain the plan for performing the test policy and strategy process.*

#### **Elaboration**

The plan for performing the test policy and strategy process can be included (or referenced by) the organization's quality plan or test process improvement plan.

**GP 2.3 Provide resources**

*Provide adequate resources for performing the test policy and strategy process, developing the test work products, and providing the services of the process.*

**GP 2.4 Assign responsibilities**

*Assign responsibility and authority for performing the test policy and strategy process, developing the work products, and providing the services of the test policy and strategy process.*

**Elaboration**

A group with the authority and knowledge is designated to be responsible for defining a test policy, test strategy and test performance indicators. The group typically consists of the following stakeholders: resource management, business management, quality management, project management, operations, test management and test engineers.

**GP 2.5 Train People**

*Train the people performing or supporting the test policy and strategy process as needed.*

**Elaboration**

People involved in the practices of defining and maintaining the test policy and test strategy are provided with basic knowledge regarding structured testing. Those involved in the practices around test performance indicators are trained on measurement practices.

**GP 2.6 Manage Configurations**

*Place designated work products of the test policy and strategy process under appropriate levels of configuration control.*

**Elaboration**

*Examples of work products placed under configuration management include the following:*

- Test policy
- Test strategy
- Test performance indicators
- Measurement data

**GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the test policy and strategy process as planned.*

**GP 2.8 Monitor and control the process**

*Monitor and Control the test policy and strategy process against the plan for performing the process and take appropriate actions.*

**GP 2.9 Objectively Evaluate Adherence**

*Objective evaluate adherence of the test policy and strategy process against its process description, standards, and procedures, and address non-compliances.*

**Elaboration**

*Examples of review and/or audit evaluation adherence topics include the following:*

- Compliance in test plans to test policy and test strategy
- Level of familiarity by test professionals and other stakeholders with test policy, test strategy and test performance indicators
- Test performance indicators data is gathered and results are reported to the stakeholders

**GP 2.10 Review status with higher level management**

*Review the activities, status and results of the test policy and strategy process with higher level management and resolve issues.*

**GG 3 Institutionalize a Defined Process**

*Only applicable at TMMi level 3*

**GP 3.1 Establish a defined process**

*Establish and maintain a description of a defined test policy and strategy process*

**GP 3.2 Collect improvement information**

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the test policy and strategy process to support the future use and improvement of the organization's processes and process assets.*

## PA 2.2 Test Planning

### Purpose

The purpose of “Test Planning” is to define a test approach based on the identified risks and the defined test strategy, and to establish and maintain well-founded plans for performing and managing the testing activities.

### Introductory Notes

After confirmation of the test assignment, an overall study is carried out regarding the product to be tested, the project organization, the requirements, and the development process. As part of test planning, the test approach is defined based on the outcome of a product risk assessment and derived from the test strategy. Depending on the level and type of risks, it is decided which requirements of the product will be tested, to what degree, how and when. The objective is to provide the best possible level and type of coverage to the parts of the system with the highest risk level. Based on the test approach the work to be done is estimated and as a result the proposed test approach is provided with clear cost. The product risks, test approach and estimates are defined in close co-operation with the stakeholders. Testers should not take these decisions themselves. The test plan will either confirm, or explain non-compliance, with the test strategy. Within test planning, the test deliverables that are to be provided are identified, the resources that are needed are determined, and aspects relating to infrastructure are defined. In addition, test project risks regarding testing are identified. As a result the test plan will define what testing is required, when, how and by whom. Finally, the test plan document is developed and committed upon by the stakeholders. The test plan provides the basis for performing and controlling the testing activities. The test plan will usually need to be revised, using a formal change control process, as the project progresses to address changes in the requirements and commitments, inaccurate estimates, corrective actions, and (test) process changes.

### Scope

The process area “Test Planning” involves performing a product risk assessment on the test object and defining a differentiated test approach based on the risks identified. It also involved developing estimates for the testing to be performed, establishing necessary commitments, and defining and maintaining the plan to perform and manage the testing. A test plan is required for each identified test level. At TMMi level 2 test plans are typically developed per test level.

### Specific Goal and Practice Summary

- SG1 Perform a product risk assessment
  - SP 1.1 Define product risk categories and parameters
  - SP 1.2 Identify product risks
  - SP 1.3 Analyze product risks
- SG2 Establish a test approach
  - SP 2.1 Identify items and features to be tested
  - SP 2.2 Define the test approach
  - SP 2.3 Define entry criteria
  - SP 2.4 Define exit criteria
  - SP 2.5 Define suspension and resumption criteria
- SG3 Establish test estimates
  - SP 3.1 Establish a top-level work breakdown structure
  - SP 3.2 Define test lifecycle
  - SP 3.3 Determine estimates of test effort and cost
- SG4 Develop a test plan
  - SP 4.1 Establish the test schedule
  - SP 4.2 Plan for test staffing
  - SP 4.3 Plan stakeholder involvement
  - SP 4.4 Identify test project risks
  - SP 4.5 Establish the test plan
- SG5 Obtain commitment to the test plan
  - SP 5.1 Review test plan

- SP 5.2 Reconcile work and resource levels  
SP 5.3 Obtain test plan commitments

## Specific Practices by Goals

### SG 1 Perform Product Risk Assessment

*A product risk assessment is performed to identify the critical areas for testing.*

#### SP 1.1 Define product risk categories and parameters

*Product risk categories and parameters are defined that will be used during the risk assessment.*

##### **Typical work products**

1. Product risk categories lists
2. Product risk evaluation and prioritization criteria

##### **Sub-practices**

1. Determine product risk categories

A reason for identifying product risk categories is to help in the future consolidation of the test tasks into test types in the test plans.

*Examples of product risk categories include the following:*

- Functional risks
- Architectural risks
- Non-functional risks, e.g. usability, efficiency, portability, maintainability, reliability
- Change related risks, e.g. regression

2. Define consistent criteria for evaluating and quantifying the product risk likelihood and impact levels
3. Define thresholds for each product risk level

Risk level is defined as the importance of a risk as defined by its characteristics impact and likelihood. For each risk level, thresholds can be established to determine the acceptability or unacceptability of product risk, prioritization of product risks, or trigger of management action.

#### SP 1.2 Identify product risks

*Product risks are identified and documented.*

##### **Typical work products**

1. Identified product risks

##### **Sub-practices**

1. Identify and select stakeholders that need to contribute to the risk assessment
2. Identify product risks using input from stakeholders and requirements documents

*Examples of product risk identification techniques include the following:*

- Risk workshops
- Brainstorming
- Expert interviews
- Checklists
- Lessons learned

3. Document the background and potential consequences of the risk
4. Identify the relevant stakeholders associated for each risk
5. Review the identified product risks against the test assignment

#### SP 1.3 Analyze product risks

*Product risks are evaluated, categorized and prioritized using predefined categories and parameters.*

##### **Typical work products**

1. Product risk list, with a category and priority assigned to each risk

**Sub-practices**

1. Analyze the identified products risks using the predefined parameters, e.g. likelihood and impact
2. Categorize and group product risks according to the defined risk categories
3. Prioritize the product risks for mitigation
4. Establish a horizontal traceability between products risks and requirements to ensure that the source of product risks is documented
5. Generate a requirements / product risks traceability matrix
6. Review and obtain agreement with stakeholders on the completeness, category and priority level of the product risks
7. Revise the product risks as appropriate

*Examples of when product risks may need to be revised include the following:*

- New or changing requirements
- Change of the software development approach
- Lessons learned on quality issues in the project

**SG 2 Establish a Test Approach**

*A test approach, based on identified product risks, is established and agreed upon.*

**SP 2.1 Identify items and features to be tested**

*The items and features to be tested, and not to be tested, are identified based on the product risks.*

**Typical work products**

1. List of items to be tested and not to be tested
2. List of features to be tested and not to be tested

**Sub-practices**

1. Breakdown the prioritized product risks into items to be tested and not to be tested
2. Document the risk level and source documentation (test basis) for each identified item to be tested
3. Breakdown the prioritized product risks into features to be tested and not to be tested
4. Document the risk level and source documentation (test basis) for each identified feature to be tested
5. Review the list of item and feature to be tested, and not to be tested with stakeholders

**SP 2.2 Define the test approach**

*The test approach is defined to mitigate the identified and prioritized product risks.*

**Typical work products**

1. Test approach
2. The approach, e.g. selected set of test design techniques, should be described in sufficient detail to support at a later stage identification of major test tasks and estimation of the time required to do each one.

**Sub-practices**

1. Select the test design techniques to be used; multiple test design techniques are defined to vary in test coverage based on the defined product risks

*Criteria for selecting a test design technique include the following:*

- Type of system
- Regulatory standards
- Customer or contractual requirements
- Level of risk
- Type of risk,
- Documentation available,
- Knowledge of the testers,
- Time and budget,

- Development lifecycle,
  - Previous experience of types of defects found
2. Define the approach to review test work products
  3. Define the approach for re-testing
    - Examples of an approach for re-testing include the following:*
      - For all high risk test items a full re-test will take place re-executing the full test procedure
      - For all low risk test items the incidents are re-tested in isolation tools
  4. Define the approach for regression testing
    - Examples of elements of a regression test approach include the following:*
      - Focus of the regression testing, e.g. which items and/or features
      - Test case to be selected and executed
      - Type of testing to be performed
      - Manual testing or using test automation tools
  5. Identify the supporting test tools to be used
  6. Identify significant constraints regarding the test approach
    - Examples of constraints regarding the test approach include the following:*
      - Test resources availability
      - Test environment features
      - Project deadlines
  7. Align the test approach with the defined organization-wide or program-wide test strategy
  8. Identify any non-compliances to the test strategy and its rationale
  9. Review the test approach with stakeholders
  10. Revise the test approach as appropriate
    - Examples of when the test approach may need to be revised include the following:*
      - New or changed priority level of product risks
      - Lessons learned on applying the test approach in the project

### SP 2.3 Define entry criteria

The entry criteria for testing are defined to prevent testing from starting under conditions that do not allow for a thorough test process.

#### Typical work products

1. Entry criteria

#### Sub-practices

1. Define a set of entry criteria related to the test process

*Examples of entry criteria related to the test process include the following:*

- The availability of a test summary report from the previous test level
- The availability of a test environment according to requirements
- The availability of documentation, e.g. test release note, user manual, installation manual, etc

2. Define a set of entry criteria related to product quality

*Examples of entry criteria related to product quality include the following:*

- A successful intake test
- No outstanding defects (of priority level X)
- All outstanding defects have at least the status analyzed

3. Review the entry criteria with stakeholders, especially those responsible for meeting the entry criteria

### SP 2.4 Define exit criteria

The exit criteria for testing are defined to plan when to stop testing.



### Typical work products

1. Exit criteria

### Sub-practices

1. Define a set of exit criteria related to the test process

*Examples of exit criteria related to the test process include the following:*

- Percentage of tests prepared that have been executed (successfully)
- Percentage of coverage for each test item, e.g. code coverage or requirements coverage
- The availability of an approved test summary report

2. Define a set of exit criteria related to product quality

*Examples of exit criteria related to product quality include the following:*

- All high priority product risks mitigated
- Defect detection rate falls below a threshold
- Number of outstanding defects (by priority level)
- Percentage of software modules supported by an inspected design

3. Review the exit criteria with stakeholders

Note that the exit criteria of a test level should be aligned with the entry criteria of a subsequent test level.

## SP 2.5 Define suspension and resumption criteria

*Criteria are defined used to suspend all or a portion of the test tasks on the test items and/or features.*

### Typical work products

1. Suspension criteria
2. Resumption criteria

### Sub-practices

1. Specify the suspension criteria used to suspend all or a portion of the test tasks on the test items and/or features

*Examples of suspension criteria include the following:*

- Number of critical defects
- Number of non reproducible defects
- Issues with test execution due to the test environments

2. Specify the resumption criteria used to specify the test tasks that must be repeated

## SG 3 Establish Test Estimates

*Well founded test estimates are established and maintained for use in discussing the test approach and in planning the testing activities.*

### SP 3.1 Establish a top-level work breakdown structure

*Establish a top-level work breakdown structure (WBS) to clearly define the scope of the test estimate.*

### Typical work products

1. Test work products list
2. Test tasks to be performed
3. Work breakdown structure

### Sub-practices

1. Identify test work products to be developed based on the defined test approach
2. Identify test work products that will be externally acquired
3. Identify test work products that will be re-used
4. Identify test tasks to be performed related to the test work products
5. Identify indirect test tasks to be performed such as test management, meetings, configuration management, etc.



Note that the WBS should also take into account tasks for implementing the test environment requirements. Refer to the “Test Environment” process area for more information.

### SP 3.2 Define test lifecycle

*Define the test lifecycle phases on which to scope the planning effort.*

#### **Typical work products**

1. Test lifecycle phases
2. Test milestones

#### **Sub-practices**

1. Define test lifecycle phases; at a minimum a test planning, test preparation and test execution phase are distinguished
2. Schedule the test preparation phase such that it starts immediately upon the completion of the test basis
3. Align the top-level work breakdown structure with the defined test lifecycle
4. Identify major milestones for each test lifecycle phase

Note that understanding the lifecycle is crucial in determining the scope of the test planning effort and the timing of the initial planning, as well as the timing and criteria (at critical milestones) for re-planning.

### SP 3.3 Determine estimates of test effort and cost

*Estimate the test effort and cost for the test work products and tasks based on estimation rationale.*

#### **Typical work products**

1. Attribute estimates of test work products and test tasks
2. Test effort estimates
3. Test cost estimates

#### **Sub-practices**

1. Determine and maintain estimates of the attributes of the test work products and test tasks

*Examples of attributes used to estimate test work products and test tasks include the following:*

- Size, e.g. number of test cases, number of pages, number of test points, volume of test data, number of requirements
- Complexity of related test item, e.g. cyclomatic number
- Level of re-use
- Priority level of related product risk

Note that appropriate methods (e.g. validated models or historical data) should be used to determine the attributes of the test work products and test tasks that will be used to estimate the resource requirements.

2. Study (technical) factors that can influence the test estimate

*Examples of factors that can influence the test estimate include the following:*

- Usage of test tools
- Quality of earlier test levels
- Quality of test basis
- Development environment
- Test environment
- Test ware availability from previous projects
- Knowledge and skill level of testers

3. Select models and/or historical data that will be used to transform the attributes of the test work products and test tasks into estimates of the effort and cost

*Examples of models that can be used for test estimation include the following:*

- Test Point Analysis [TMap]
- Three point estimate
- Wide Band Delphi [Veenendaal]

- Ratio on development effort versus test effort
4. Include supporting infrastructure needs when estimation test effort and cost

*Examples of supporting infrastructure needs include the following:*

    - Test environment
    - Critical computer resources
    - Office environment
    - Test tools
  5. Estimate test effort and cost using models and/or historical data
  6. Document assumptions made in deriving the estimates
  7. Record the test estimation data, including the associated information needed to reconstruct the estimates

## SG 4 Develop a Test Plan

*A test plan is established and maintained as the basis for managing testing and communication to stakeholders.*

### SP 4.1 Establish the test schedule

*The test schedule with predefined stages of manageable size is established and maintained based on the developed test estimate and defined test lifecycle.*

#### **Typical work products**

1. Test schedule

#### **Sub-practices**

1. Identify test scheduling constraints such as task duration, resources, and inputs needed
2. Identify test task dependencies
3. Define the test schedule (timing of testing activities, test lifecycle phases and test milestones)
4. Document assumptions made in defining the test schedule
5. Establish corrective action criteria for determining what constitutes a significant deviation from the test plan and may require rescheduling.

### SP 4.2 Plan for test staffing

*Plan for necessary test staffing resources with the required knowledge and skills to perform the testing.*

#### **Typical work products**

1. Staffing requirements
2. Inventory of skill needs
3. Staffing and new hire plan
4. Test training plan

#### **Sub-practices**

1. Determine staffing requirements based on the work breakdown structure, test estimate and test schedule
2. Identify knowledge and skills needed to perform the test tasks
3. Assess the knowledge and skills available
4. Select mechanisms for providing needed knowledge and skills

*Examples of mechanisms include the following:*

- In-house training
  - External training
  - Coaching
  - External skill acquisition
5. Incorporate selected mechanisms into the test plan

**SP 4.3 Plan stakeholder involvement**

*Plan the involvement of identified stakeholders.*

Stakeholders are identified from all phase of the test lifecycle by identifying the type of people and functions needing, representation in testing activities and describing their relevance and the degree of interaction for specific testing activities. A two-dimensional matrix with stakeholders along one axis and testing activities along the other axis is convenient for accomplishing this identification.

**Typical work products**

1. Stakeholder involvement plan

**SP 4.4 Identify test project risks**

*The test project risks associated with testing are identified, analyzed and documented.*

**Typical work products**

1. Identified test project risks
2. Prioritized test project risk list
3. Test project risk contingencies

**Sub-practices**

1. Identify test project risks

*Examples of project risk identification techniques include the following:*

- Brainstorming
- Expert interviews
- Checklists

2. Analyze the identified test project risks in terms of likelihood and impact
3. Prioritize the analyzed test project risks
4. Review and obtain agreement with stakeholders on the completeness and priority level of the documented test project risks
5. Define contingencies for the (high priority) test project risks
6. Revise the test project risks as appropriate

*Examples of when test project risks may need to be revised include:*

- When new test project risks are identified
- When the likelihood of a test project risk changes
- When test project risks are retired
- When testing circumstances change significantly

**SP 4.5 Establish the test plan**

*The test plan is established and maintained as a basis for managing testing.*

The results of previous practices are documented in an overall test plan, tying together the information in a logical manner.

**Typical work products**

1. Test Plan

*Examples of elements of a test plan include the following [after IEEE 829]:*

- Test plan identifier
- An overall introduction
- Non-compliances to the test strategy and its rationale
- Items to be tested (including priority level) and not to be tested
- Features to be tested (including priority level) and not to be tested
- Test approach (e.g. test design techniques)
- Entry and exit criteria
- Suspension and resumption criteria
- Test milestones and work products

- Test lifecycle and tasks
- Environmental needs and requirements (including office environment)
- Staffing and training needs
- Stakeholder involvement
- Test estimate
- Test schedule
- Test project risks and contingencies

Refer to the “Test Environment” process area for information on environmental needs and requirements.

## **SG 5 Obtain Commitment to the Test Plan**

*Commitments to the test plan are established and maintained.*

### **SP 5.1 Review test plan**

*Review the test plan (and possibly other plans) that affect testing to understand test commitments.*

#### **Typical work products**

1. Test plan review log

#### **Sub-practices**

1. Organize reviews with stakeholders to make them understand test commitments

### **SP 5.2 Reconcile work and resource levels**

*Review the test plan to reflect available and estimated resources.*

#### **Typical work products**

1. Revised test approach and corresponding estimation parameters
2. Renegotiated test budgets
3. Revised test schedules
4. Revised product risk list
5. Renegotiated stakeholder agreements

#### **Sub-practices**

1. Discuss differences between estimates and available resources with stakeholders
2. Reconcile any differences between estimates and available resources

Note that reconciliation is typically accomplished by lowering or deferring technical performance, negotiating more resources, finding ways to increase productivity, outsourcing, adjusting staff skill mix, or revising the schedule.

### **SP 5.3 Obtain test plan commitments**

*Obtain commitments from relevant stakeholders responsible for performing and supporting test plan execution.*

#### **Typical work products**

1. Documented requests for commitments
2. Documented commitments

#### **Sub-practices**

3. Identify needed support and negotiate commitments with relevant stakeholders

Note that the WBS can be used as a checklist for ensuring that commitments are obtained for all tasks. The plan for stakeholders' interaction should identify all parties from whom commitments should be obtained.

1. Document all organizational commitments, both full and provisional
2. Review internal commitments with senior management as appropriate
3. Review external commitments with senior management as appropriate

## Generic Practices by Goals

### GG 2 Institutionalize a Managed Process

#### GP 2.1 Establish an organizational policy

*Establish and maintain an organizational policy for planning and performing the test planning process.*

##### **Elaboration**

The test planning policy typically specifies:

- Each project defines a test plan that includes a test approach and the accompanying test effort and estimates
- The project's test approach is derived from the test strategy
- Test plans shall be developed using a standard process and template
- Standard tools that will be used when performing test planning
- The requirements are used as a basis for test planning activities
- The testing commitments are negotiated with resource management, business management and project management
- Involvement of other affected groups in the testing activities is explicitly agreed upon by these groups
- Management explicitly reviews all testing commitments made to groups external to the organization
- The test plan is managed and controlled

#### GP 2.2 Plan the process

*Establish and maintain the plan for performing the test planning process.*

##### **Elaboration**

Typically, the plan for performing the test planning process is included in the project plan, which is described in the CMMI Project Planning process area.

#### GP 2.3 Provide resources

*Provide adequate resources for performing the test planning process, developing the test work products, and providing the services of the process.*

##### **Elaboration**

- A documented and approved assignment exists for testing typically covering issues and expectation regarding goals and objectives, exit criteria, items and features to be tested and not to be tested, type of testing to be performed, imposed standards, cost, schedule and resource constraints.
- Adequate time is provided to test management to perform the test planning activities
- Experienced individuals, who have expertise in the application domain of the test object and those who have expertise on the development process are available to support the development of the test plan
- Tools to support the test planning process are available

*Examples of tools include the following:*

- Project planning and scheduling tools
- Estimation tools
- Risk assessment tools
- Test management tools
- Configuration management

#### GP 2.4 Assign responsibilities

*Assign responsibility and authority for performing the test planning process, developing the work products, and providing the services of the test planning process.*

**Elaboration**

A test manager is typically designated to be responsible for negotiating commitments and developing the test plan. The test manager, directly or by delegation, co-ordinates the project's test planning process.

**GP 2.5 Train people**

*Train the people performing or supporting the test planning process as needed.*

**Elaboration**

Test management, and other individuals or groups, involved in test planning, are trained in test planning and the accompanying procedures and techniques.

*Examples of topics include the following:*

- Planning principles
- Test strategy
- Product and project risk assessment process and techniques
- Defining a test approach
- Test plan templates and standards
- Test organization
- Test estimation and test scheduling
- Introduction on test design techniques
- Supporting test planning tools

**GP 2.6 Manage configurations**

*Place designated work products of the test planning process under appropriate levels of configuration control.*

**Elaboration**

*Examples of work products placed under configuration management include the following:*

- Work breakdown structure
- Test estimation data
- Product risk assessment data
- Test plan review report
- Test plan

**GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the test planning process as planned.*

**Elaboration**

Select relevant stakeholders from customers, end users, developers, producers, testers, suppliers, marketers, maintainers, service personnel, and others who may be affected by, or may affect, the product as well as the test process.

*Examples of activities for stakeholder involvement include the following:*

- Selecting the product and product components to be tested
- Identifying the risk level and risk types of the product and product components to be tested by being involved in a product risk assessment
- Providing input to test estimates
- Reviewing and resolving issues on test project risks
- Explicitly committing to test resources needed
- Reviewing and approval of test plan

**GP 2.8 Monitor and control the process**

Monitor and Control the test planning process against the plan for performing the process and take appropriate actions.

**Elaboration**

*Examples of measures used in monitoring and control the test planning process include the following:*

- Number of revisions to the test plan
- Effort spent and actual lead-time compared to the effort planned and planned lead-time in the test plan
- Number of test items' risk level changed per revision
- Cost, schedule and effort variance per plan revision

**GP 2.9 Objectively evaluate adherence**

*Objective evaluate adherence of the test planning process against its process description, standards, and procedures, and address non-compliances.*

**Elaboration**

*Examples of review and/or audit evaluation adherence topics include the following:*

- Compliance to the test strategy
- Compliance to standards (procedures and templates)
- The quality of the test plan
- The defined test approach
- The risk assessment process
- The test estimation process
- The activities for reviewing and making test commitments

**GP 2.10 Review status with higher level management**

*Review the activities, status and results of the test planning process with higher level management and resolve issues.*

**GG 3 Institutionalize a Defined Process**

*Only applicable at TMMi level 3*

**GP 3.1 Establish a defined process**

*Establish and maintain a description of a defined test planning process*

**GP 3.2 Collect improvement information**

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the test planning process to support the future use and improvement of the organization's processes and process assets.*

*Examples of measures include the following:*

- Percentage of test plans established according to procedure and template
- Percentage of test plans that have documented product risk assessment results and a test approach
- Percentage of test plans formally reviewed and approved by management
- Test planning effort
- Test estimation accuracy



## PA 2.3 Test Monitoring and Control

### Purpose

The purpose of Test Monitoring and Control is to provide an understanding of test progress and product quality so that appropriate corrective actions can be taken when test progress deviates significantly from plan or product quality deviates significantly from expectations.

### Introductory Notes

The progress of testing and the quality of the products should both be monitored and controlled. The progress of the testing is monitored by comparing the status of actual test (work) products, tasks (including their attributes), effort, cost, and schedule to the test plan. The quality of the product is monitored by means of indicators such as product risks mitigated, the number of defects found, number of open defects, and status against test exit criteria. Monitoring involves gathering the required (raw) data, e.g. from test log and test incidents reports, reviewing the raw data on their validity and calculating the defined progress and product quality measures. Test summary reports should be written on a periodic and event driven basis as a means to provide a common understanding on test progress and product quality. Since 'testing is the measurement of product quality' [Hetzel], especially the practices around product quality reporting are key to the success of this process area. Appropriate corrective actions should be taken in time when the test progress deviates from the plan or product quality deviates from expectations. These actions may require re-planning, which may include revising the original plan or additional mitigations activities with the current plan. Corrective actions that influence the original committed plan should be agreed with the stakeholders.

An essential part of test monitoring and control is test project risk management. Test project risk management is performed to identify and solve major problems that undermine the test plan as early as possible. When performing project risk management, it is also important to identify problems that are beyond the responsibility of testing. For instance, organizational budget cuts, delay of development work products or changed/added functionality can all affect the test process significantly. Building on the test projects risks documented in the test plan, test project risks are monitored and controlled and corrective actions are initiated as needed.

### Scope

The process area "Test Monitoring and Control" involves monitoring the test progress and product quality against documented estimates, commitments, plans and expectations, reporting on test progress and product quality to stakeholders, taking control measures, e.g. corrective actions, when necessary and managing the corrective actions to closure.

### Specific Goal and Practice Summary

- SG1 Monitor test progress against plan
  - SP 1.1 Monitor test planning parameters
  - SP 1.2 Monitor test environment resources provided and used
  - SP 1.3 Monitor test commitments
  - SP 1.4 Monitor test project risks
  - SP 1.5 Monitor stakeholder involvement
  - SP 1.6 Conduct test progress reviews
  - SP 1.7 Conduct test progress milestone reviews
- SG2 Monitor product quality against plan and expectations
  - SP 2.1 Check against entry criteria
  - SP 2.2 Monitor defects
  - SP 2.3 Monitor product risks
  - SP 2.4 Monitor exit criteria
  - SP 2.5 Monitor suspension and resumption criteria
  - SP 2.6 Conduct product quality reviews
  - SP 2.7 Conduct product quality milestone reviews
- SG3 Manage corrective action to closure
  - SP 3.1 Analyze issues
  - SP 3.2 Take corrective action



## Specific Practices by Goals

### SG 1 Monitor Test Progress Against Plan

*Actual progress and performance of testing is monitored against the test plan.*

#### SP 1.1 Monitor test planning parameters

Monitor the actual values of test planning parameters against the test plan.

##### **Typical work products**

1. Records of test performance
2. Records of significant deviations from plan

##### **Sub-practices**

1. Monitor test progress against the test schedule

*Examples of progress monitoring typically include the following:*

- Periodically measuring the actual completion of test tasks, test (work) products and test milestones
- Comparing actual completion of test tasks, test (work) products and test milestones against the test schedule documented in the test plan
- Identifying significant deviations from the test schedule estimates in the test plan

2. Monitor the test cost and expended test effort

*Examples of cost and effort monitoring typically include the following:*

- Periodically measuring the actual test costs and effort expended and staff assigned
- Comparing actual test cost, effort and staffing to the estimates documented in the test plan
- Identifying significant deviations from the test cost, effort and staffing in the test plan

3. Monitor the attributes of the test work products and test tasks

Refer SP 3.3 "Determine estimates of test effort and cost" from the "Test Planning" process area for information about the attributes of test work products and test tasks.

*Examples of test work products and test tasks attributes monitoring typically include the following:*

- Periodically measuring the actual attributes of the test work products and test tasks, such as size or complexity
- Comparing the actual attributes of the test work products and test tasks to the estimates documented in the test plan
- Identifying significant deviations from the estimated in the test plan

4. Monitor the knowledge and skills of test staffing

*Examples of knowledge and skills monitoring typically include the following:*

- Periodically measuring the acquisition of knowledge and skills of test staffing
- Comparing actual training obtained to that documented in the test plan
- Identifying significant deviations from estimates in the test plan
- The ratio of defects being solved versus defects being found

5. Document the significant deviations in the test planning parameters.

#### SP 1.2 Monitor test environment resources provided and used

*Monitor the test environments resources provided and used against the plan.*

##### **Typical work products**

1. Records of test environment resources provided and used
2. Records of significant deviations from plan

##### **Sub-practices**

1. Monitor test environments resources provided against plan
2. Monitor the actual usage of the provided test environments resources against plan

3. Identify and document significant deviations from the estimates in the plan

### SP 1.3 Monitor test commitments

*Monitor test commitments against those identified in the test plan.*

#### **Typical work products**

1. Records of commitment reviews

#### **Sub-practices**

1. Regularly review commitments (both internal and external)
2. Identify commitments that have not been satisfied or that are at significant risk of not being satisfied
3. Document the results of the commitment reviews.

### SP 1.4 Monitor test project risks

*Monitor test project risks against those identified in the test plan*

#### **Typical work products**

1. Updated test project risk list
2. Records of project risk monitoring

#### **Sub-practices**

1. Periodically review the documentation of the test project risks in the context of the current status and circumstances
2. Revise the documentation of the test project risks, as additional information becomes available, to incorporate the change
3. Communicate test project risk status to relevant stakeholders

### SP 1.5 Monitor stakeholder involvement

*Monitor stakeholder involvement against the test plan.*

Once the stakeholders are identified and the extent of their involvement within testing is specified in the test plan, that involvement must be monitored to ensure that the appropriate interactions are occurring.

#### **Typical work products**

1. Records of stakeholder involvement

#### **Sub-practices**

1. Periodically review the status of stakeholder involvement
2. Identify and document significant issues and their impact
3. Document the results of the stakeholder involvement status reviews

### SP 1.6 Conduct test progress reviews

*Periodically review test progress, performance and issues*

Progress reviews are reviews to keep stakeholders informed. Reviews are often held both internally with test team members and with stakeholders outside testing. These reviews are typically informal reviews held regularly, e.g. weekly, bi-weekly or monthly.

#### **Typical work products**

1. Test progress report
2. Documented test progress review results, e.g. minutes of the progress meetings

#### **Sub-practices**

1. Collect and analyze test progress monitoring measures
2. Regularly communicate status on test progress and performance to stakeholders

*Examples of stakeholders typically include the following:*

- Project management
- Business management

- Test team members

3. Regularly organize test progress review meetings with stakeholders
4. Identify, document and discuss significant issues and deviations for the test plan
5. Document change requests on test work products and major problems identified in test progress and performance
6. Document the results of the reviews, e.g. decisions taken

### SP 1.7 Conduct test progress milestone reviews

*Review the accomplishments and progress of testing at selected test milestones*

Test progress milestone reviews are planned during test planning and are typically formal reviews.

#### **Typical work products**

1. Test milestone report
2. Documented milestone review results, e.g. minutes of the review meeting

#### **Sub-practices**

1. Conduct test progress reviews at meaningful points in the test schedule, such as the completion of selected stages, with relevant stakeholders
2. Communicate accomplishments and test progress and performance status to stakeholders
3. Review the commitments, plan, status, and project risks of testing
4. Review the test environment resources
5. Identify, document and discuss significant test progress issues and their impacts
6. Document the results of the reviews, actions items, and decisions
7. Update the test plan to reflect accomplishments and latest status

## SG 2 Monitor Product Quality Against Plan And Expectations

*Actual product quality is monitored against plan and expectations.*

### SP 2.1 Check against entry criteria

*At the start of the test execution phase check the status against the entry criteria identified in the test plan.*

#### **Typical work products**

1. Records of entry check

#### **Sub-practices**

1. Check the status against the entry criteria identified in the test plan
2. Identify and document significant deviations in compliance to entry criteria and initiate corrective action

### SP 2.2 Monitor defects

*Monitor measures on defects found during testing against expectations.*

#### **Typical work products**

1. Records of defect monitoring

#### **Sub-practices**

1. Monitor measures on defects found and status against expectations

*Examples of useful defect measures include the following [Burnstein]:*

- Total number of defects (for a component, subsystem, system) outstanding at each defined priority level
- Total number of defects found during the most recent test run at each defined priority level
- Number of defects resolved/unresolved (for all levels of test)
- Number of defects found for each given type
- Number of defects causing failures of severity level greater than X

- Number of defects/KLOC (“incident volume”)
  - Actual number versus estimated number of defects (based on historical data)
  -
2. Identify and document significant deviations on measures regarding defects found from expectations

### SP 2.3 Monitor product risks

*Monitor product risks against those identified in the test plan.*

#### **Typical work products**

1. Updated test product risk list
2. Records of product risk monitoring

#### **Sub-practices**

1. Periodically review the documentation of the product risks in the context of the current status and circumstances with a selected set of stakeholders
2. Monitor changes and additions to the requirements to identify new or changed products risks
3. Revise the documentation of the product risks, as additional information becomes available to incorporate the change on likelihood, impact and/or priority status
4. Monitor the (number of) product risks mitigated by testing against plan
5. Communicate product risks status to relevant stakeholders

### SP 2.4 Monitor exit criteria

*Monitor the status of the exit criteria against those identified in the test plan.*

#### **Typical work products**

1. Records of exit criteria monitoring

#### **Sub-practices**

1. Monitor the test process related exit criteria, e.g. test coverage against plan
2. Monitor the product quality related exit criteria against plan
3. Identify and document significant deviations in exit criteria status from plan

### SP 2.5 Monitor suspension and resumption criteria

*Monitor the status of the suspension and resumption criteria against those identified in the test plan.*

#### **Typical work products**

1. Records of suspension criteria monitoring
2. Records of resumption criteria monitoring

#### **Sub-practices**

1. Monitor suspension criteria against those documented in the test plan
2. Suspend testing if suspension criteria are met and initiate corrective action
3. Monitor resumption criteria against those documented in the test plan
4. Initiate the resumption of testing once the issues have been solved using the defined resumption criteria

### SP 2.6 Conduct product quality reviews

*Periodically review product quality*

Product quality reviews are reviews to keep stakeholders informed. Reviews are often held both internally with test team members and with stakeholders outside testing. These reviews are typically informal reviews held regularly, e.g. weekly, bi-weekly or monthly.

#### **Typical work products**

1. Product quality report
2. Documented product quality review results, e.g. minutes of the product quality meetings

### Sub-practices

1. Collect and analyze product quality monitoring measures
2. Regularly communicate status on product quality to stakeholders

*Examples of stakeholders typically include the following:*

- Project management
- Business management
- Test team members

3. Regularly organize product quality review meetings with stakeholders
4. Identify, document and discuss significant product quality issues and deviations from expectations and plan
5. Document the results of the reviews, e.g. decisions taken

## SP 2.7

### Conduct product quality milestone reviews

*Review product quality status at selected test milestones*

Product quality milestone reviews are planned during test planning and are typically formal reviews.

#### Typical work products

1. Test milestone report
2. Documented milestone review results, e.g. minutes of the review meeting

### Sub-practices

1. Conduct product quality reviews at meaningful points in the test schedule, such as the completion of selected stages, with relevant stakeholders
2. Communicate product quality status to stakeholders by means of a formal product quality report

*Examples of elements of a product quality test report include the following [after IEEE 829]:*

- Identifier (and reference to test plan)
- Management summary
- Variances (against plan)
- Comprehensive assessment
- Summary of results
- Evaluation
- Summary of activities
- Approvals

3. Review the status regarding incidents, product risks and exit criteria
4. Identify and document significant product quality issues and their impacts
5. Document the results of the reviews, actions items, and decisions
6. Update the test plan to reflect accomplishments and the latest status

## SG 3

### Manage Corrective Actions To Closure

*Corrective actions are managed to closure when test progress or product quality deviate significantly from test plan or expectations.*

## SP 3.1

### Analyze issues

*Collect and analyze the issues and determine the corrective actions necessary to address the issues.*

#### Typical work products

1. List of issues needed corrective actions

### Sub-practices

1. Gather issues for analysis

*Examples of issues to be gathered include the following:*

- Significant deviations in test planning parameters from estimates in the test plan
- Commitments that have not been satisfied
- Significant changes in test project risk status, e.g. possible late delivery and/or poor

quality of test basis and/or test object

- Stakeholder representation or involvement issues
- Significant deviations in test environment implementation progress from plan
- Number, severity and priority level of incidents found
- Status regarding exit criteria
- Significant changes in product risks

2. Analyze issues to determine need for corrective action

Note corrective action is required when the issue, if left unresolved, may prevent testing or even the project from meeting its objectives.

### SP 3.2 Take corrective action

*Take corrective action on identified issues.*

#### **Typical work products**

1. Corrective action plan

#### **Sub-practices**

1. Determine and document the appropriate actions needed to address the identified issues

*Examples of potential actions include the following:*

- Re-negotiating commitments
- Adding resources
- Changing the test approach
- Re-visiting the exit criteria
- Deferring release date
- Changing the scope of the project, e.g. delivering less functionality

Note that many of the potential actions listed above will lead in a revised test plan.

2. Review and get agreement with relevant stakeholders on the actions to be taken
3. Re-negotiate commitments with stakeholders (both internally and externally)

### SP 3.3 Manage corrective action

*Manage corrective action to closure.*

#### **Typical work products**

1. Corrective action results

#### **Sub-practices**

1. Monitor corrective actions for completion
2. Analyze results of corrective actions to determine the effectiveness of the corrective actions

## Generic Practices by Goals

### GG 2 Institutionalize a Managed Process

#### GP 2.1 Establish an Organizational Policy

*Establish and maintain an organizational policy for planning and performing the test monitoring and control process.*

#### **Elaboration**

The test monitoring and control policy typically specifies:

- A documented test plan is used and maintained as the basis for monitoring the test progress
- Monitoring is performed on the basis of a set of test related measurement
- Test project tasks, effort and costs are monitored throughout the project
- Contingency plans are developed based on the project risks identified
- Management and other stakeholders are kept informed regarding test progress

- Management and other stakeholders are kept informed regarding product quality
- Corrective actions are taken and managed to closure when test progress deviates significantly from plan or product quality deviate significantly from expectations
- Major changes to the test plan are reviewed by management and other stakeholders

## GP 2.2 Plan the Process

*Establish and maintain the plan for performing the test monitoring and control process.*

### **Elaboration**

Typically, the plan for performing the test monitoring and control process is included in the test plan, which is described in the TMMi Test Planning process area.

## GP 2.3 Provide Resources

*Provide adequate resources for performing the test monitoring and control process, developing the test work products, and providing the services of the process.*

### **Elaboration**

- A test plan against which testing can be monitored and controlled
- Adequate time is provided to test management to perform the test monitoring and control activities
- Tools to support the test monitoring and control process are available

*Examples of tools include the following:*

- Project management and progress tracking tools
- Risk management tools
- Incident management tools
- Test management tools

## GP 2.4 Assign Responsibilities

*Assign responsibility and authority for performing the test monitoring and control process, developing the work products, and providing the services of the test monitoring and control process.*

### **Elaboration**

A test manager is typically designated to be responsible for test monitoring and control. The test manager, directly or by delegation, co-ordinates the project's test process.

*Examples of monitor and control responsibilities to be assigned include the following:*

- Monitor and control the costs, effort and schedule of testing
- Monitors and control test project risks
- Monitors and control product risks and product quality
- Reports on test progress and product quality
- Initiate corrective actions when test progress deviates significantly from test plan
- Initiate corrective actions when product quality deviates significantly from expectations

## GP 2.5 Train People

*Train the people performing or supporting the test monitoring and control process as needed.*

### **Elaboration**

Test management, and other individuals or groups, involved in test monitoring and control, are trained in test monitoring and control and the accompanying procedures and techniques.

*Examples of training topics include the following:*

- Project management fundamentals
- Managing testing
- Tracking of product quality, effort, cost and schedule
- Risk management
- Test reporting



- Contingency planning

## GP 2.6 Manage Configurations

Place designated work products of the test monitoring and control process under appropriate levels of configuration control.

### Elaboration

Examples of work products placed under configuration management include the following:

- Test schedule with status
- Test measurement data and analysis
- Test reports

## GP 2.7 Identify and Involve Relevant Stakeholders

Identify and involve relevant stakeholders of the test monitoring and control process as planned.

### Elaboration

Examples of activities for stakeholder involvement include the following:

- Assessing the testing performance against the test plan
- Reviewing commitments and resolving issues
- Reviewing product and test project risks
- Reviewing test data management activities
- Reviewing test progress and product quality
- Managing corrective actions to closure

Note that this generic practice only covers the involvement of relevant stakeholders in test monitoring and controlling.

## GP 2.8 Monitor and Control the Process

Monitor and Control the test monitoring and control process against the plan for performing the process and take appropriate actions.

### Elaboration

Examples of measures used in monitoring and control the test monitoring and control process include the following:

- Number of open and closed corrective actions
- Number of types of peer reviews performed
- Review schedule (planned versus actual and slipped target dates)

Note that this generic practice only covers the monitoring and controlling of test monitoring and control activities.

## GP 2.9 Objectively Evaluate Adherence

Objectively evaluate adherence of the test monitoring and control process against its process description, standards, and procedures, and address non-compliances.

### Elaboration

Examples of review and/or audit evaluation adherence topics include the following:

- The monitoring of test progress against the test plan
- Managing corrective actions to closure
- The performance of test project risk management
- Compliance to standards (procedures and templates)
- Test and quality reports
- Review results

**GP 2.10 Review status with Higher Level Management**

*Review the activities, status and results of the test monitoring and control process with higher level management and resolve issues.*

**GG 3 Institutionalize a Defined Process**

*Only applicable at TMMi level 3*

**GP 3.1 Establish a defined process**

*Establish and maintain a description of a defined test monitoring and control process*

**GP 3.2 Collect improvement information**

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the test monitoring and control process to support the future use and improvement of the organization's processes and process assets.*

*Examples of measures include the following:*

- Percentage of projects using the test reporting template
- Percentage of test milestones passed by means of a formally review
- Percentage of corrective actions closed within X days

## PA 2.4 Test Design and Execution

### Purpose

The purpose of Test Design and Execution is to improve test process capability during test design and execution by establishing test design specifications using test design techniques, performing a structured test execution process and managing test incidents to closure.

### Introductory Notes

Structured testing implies that test design techniques are applied, possibly supported by tools. Test design techniques are used to derive and select test conditions and subsequently test cases from requirements and design specifications. The test conditions and test cases are documented in a test specification. A test case consists of the description of the input values, execution preconditions, expected results and execution post conditions. At a later stage, as more information becomes available regarding the implementation, the test cases are translated into test procedures. In a test procedure, also referred to as a manual test script, the specific test actions and checks are arranged in an executable sequence. Specific test data required to be able to run the test procedure is created. The tests will subsequently be executed using these test procedures. The test design and execution activities follow the test approach as defined in the test plan. The specific test design techniques applied (e.g. black-box, white-box or experienced-based) are amongst other based on level and type of product risk identified during test planning.

During the test execution stage, incidents are found and incident reports are written. Incidents are be logged using an incident management system and communication regarding the incidents with stakeholders is established. For incident management a basic incident classification scheme is established, and a procedure is put in place to handle the incident lifecycle process and manage them to closure.

### Scope

The process area Test Design and Execution addresses the test preparation phase including the application of test design techniques to derive and select test conditions and test cases. It also addresses the creation of specific test data, the execution of the tests using documented test procedures and incident management.

### Specific Goal and Practice Summary

- SG1 Perform test analysis and design using test design techniques
  - SP 1.1 Identify and prioritize test conditions
  - SP 1.2 Identify and prioritize test cases
  - SP 1.3 Identify necessary specific test data
  - SP 1.4 Maintain horizontal traceability with requirements
- SG2 Perform test implementation
  - SP 2.1 Develop and prioritize test procedures
  - SP 2.2 Create specific test data
  - SP 2.3 Specify intake test procedure
  - SP 2.4 Develop test execution schedule
- SG3 Perform test execution
  - SP 3.1 Perform intake test
  - SP 3.2 Execute test cases
  - SP 3.3 Report test incidents
  - SP 3.4 Write test log
- SG4 Manage test incidents to closure
  - SP 4.1 Decide on test incidents in configuration control board
  - SP 4.2 Perform appropriate action to fix the test incident
  - SP 4.3 Track the status of test incidents

## Specific Practices by Goals

### SG 1 Perform Test Analysis And Design Using Test Design Techniques

*During test analysis and design the test approach is translated into tangible test conditions and test cases using test design techniques.*

#### SP 1.1 Identify and prioritize test conditions.

*Test conditions are identified and prioritized using test design techniques, based on an analysis of test items as specified in the test basis*

##### **Typical work products**

1. Test basis issue log
2. Test conditions
3. Test design specification

##### **Sub-practices**

1. Study and analyze the test basis (such as requirements, architecture, design and interface specifications)
2. Discuss issues regarding the test basis with the document owner
3. Select the most appropriate test design techniques in line with the documented test approach

*Examples of black-box test design techniques include the following:*

- Equivalence Partitioning
- Boundary Value Analysis
- Decision Tables (Cause/Effect Graphing)
- State Transition Testing

*Examples of white-box test design techniques include the following:*

- Statement Testing
- Decision (Branch) Testing
- Condition Testing

Note that in addition to black box and white box techniques also experienced-based techniques such as exploratory testing can be used which results in documenting the test design specification by means of a test charter.

Typically more than one test design technique is selected per test level in order to be able to differentiate the intensity of testing, e.g. number of test cases, based on the level of risk of the test items. In addition to risk level selection factors for test design techniques include development lifecycle, quality of the test basis, skills and knowledge of the testers, contractual requirements and imposed standards.

4. Derive the test conditions from the test basis using test design techniques
5. Prioritize the test conditions based on identified product risks
6. Document the test conditions in a test design specification, based on the test design specification standard

*Examples of elements of a test design specification include the following [after IEEE 829]:*

- Test design specification identifier
- Items and/or features to be tested
- Approach refinements
- Test conditions
- Pass/fail criteria

7. Review the test design specifications with stakeholders
8. Revise the test design specifications and test conditions as appropriate, e.g. whenever the requirements change.

#### SP 1.2 Identify and prioritize test cases

*Test cases are identified and prioritized using test design techniques*

**Typical work products**

1. Test cases
2. Test case specification

**Sub-practices**

1. Derive the test cases from the test conditions using test design techniques. A test case consists of a set of input values, execution preconditions, expected results and execution post conditions.
2. Prioritize the test cases based on identified product risks
3. Document the test cases in a test case specification, based on the test case specification standard

*Examples of elements of a test case specification include the following [IEEE 829]:*

- Test case specification identifier
  - Items and/or features to be tested
  - Input specifications
  - Output specifications
  - Environmental needs
  - Special procedural requirements
  - Inter-case dependencies
4. Review the test case specifications with stakeholders
  5. Revise the test case specifications as appropriate

**SP 1.3 Identify necessary specific test data**

*Specific test data necessary to support the test conditions and test cases is identified*

**Typical work products**

1. Test data specification

**Sub-practices**

1. Identify and specify the necessary specific test data required to implement and execute the test cases
2. Document the necessary specific test data, possibly as part of the test case specification

**SP 1.4 Maintain horizontal traceability with requirements**

*Maintain horizontal traceability from requirements to test conditions*

**Typical work products**

1. Requirements / test conditions traceability matrix

**Sub-practices**

1. Maintain requirements traceability to ensure that the source of test conditions is documented
2. Generate a requirements / test conditions traceability matrix
3. Set up the traceability matrix such that monitoring of requirements coverage during test execution is facilitated

**SG 2 Perform Test Implementation**

*Test procedures are developed and prioritized, including the intake test. Test data is created, and the test schedule is defined.*

**SP 2.1 Develop and prioritize test procedures**

*Test procedures are developed and prioritized.*

**Typical work products**

1. Test procedure specification
2. Automated test script

**Sub-practices**

1. Develop test procedures by combining the test cases in a particular order and including any other information needed for test execution
2. Prioritize the test procedures based on identified product risks
3. Document the test procedures in a test procedure specification, based on the test procedure specification standard

*Examples of elements of a test procedure specification include the following [IEEE 829]:*

- Test procedure specification identifier
  - Purpose
  - Special requirements (execution preconditions), e.g. dependencies on other test procedures
  - Procedure steps (test actions and checks)
4. Review the test procedure specifications with stakeholders
  5. Revise the test procedure specifications as appropriate
  6. Optionally, the test procedures can be automated and translated into automated test scripts

**SP 2.2 Create specific test data**

*Specific test data as specified during the test analysis and design activity is created.*

**Typical work products**

1. Test data

**Sub-practices**

1. Create specific test data required to perform the tests as specified in the test procedures
2. Archive the set of specific test data to allow a restore of the initial situation in the future

Refer to SP 3.2 “Perform test data management” from the process area “Test Environment” for managing the created test data.

**SP 2.3 Specify intake test procedure**

*The intake test (confidence test) to decide at the beginning of test execution whether the test object is ready for detailed and further testing is specified.*

**Typical work products**

1. Intake checklist
2. Intake test procedure specification

**Sub-practices**

1. Define a list of checks to be carried during the intake test using the entry criteria as defined in the test plan as an input

*Examples of checks to be part of an intake test include the following:*

- All major functions that need to be accessible
  - Representative functions that will be tested by means of a simple valid test case
  - Interfaces with other components or systems that will be tested by means of a test case to check the communication
  - Completeness of documentation, e.g. test release note, user manual, installation manual, etc.
2. Develop the intake test procedure based on the checks identified by putting the checks (test cases) in an executable order and including any other information needed for test execution
  3. Document the intake test procedures in a test procedure specification, based on the test procedure specification standard
  4. Review the intake test procedure specification with stakeholders

**SP 2.4 Develop test execution schedule**

*A test execution schedule is developed that described the sequence in with the test procedures will be executed.*

**Typical work products**

1. Test execution schedule

**Sub-practices**

1. Investigate the dependencies between the test procedures
2. Schedule the test procedures using their priority level as a main driver
3. Assign a tester to perform the execution of a test procedure
4. Review the test execution schedule with stakeholders
5. Revise the test execution schedule as appropriate

**SG 3 Perform Test Execution**

*Tests are executed in line with previously specified test procedures and test schedule. Incidents are reported and test logs are written.*

**SP 3.1 Perform intake test**

*Perform the intake test (confidence test) to decide whether the test object is ready for detailed and further testing.*

**Typical work products**

1. Intake test log
2. Incident reports

**Sub-practices**

1. Perform the intake test (confidence test) using the documented intake test procedure to decide if the test object is ready for detailed and further testing.
2. Document the results of the intake test by means of a test log, based on the test log standard
3. Log incidents when a discrepancy is observed

Note that this practice is highly related to the practice SP2.4 "Perform test environment intake test" from the process area Test Environment. The intake test on the test object and test environment can possibly be combined.

**SP 3.2 Execute test cases**

*The test cases are executed manually using documented test procedures and/or or automated using test scripts.*

**Typical work products**

1. Test results

**Sub-practices**

1. Execute the test cases using document test procedures and/or test scripts
2. Record actual results
3. Compare actual results with expected results
4. Repeat test activities as a result of an action for an incident found by performing perform re-testing (confirmation testing)
5. Perform regression testing as appropriate.

Note that some testing will be carried informally using no pre-defined detailed test procedures, e.g. during exploratory testing or error guessing.

**SP 3.3 Report test incidents**

*Discrepancies are reported as test incidents when there are differences between actual and expected results.*

**Typical work products**

1. Test incident reports

**Sub-practices**

1. Log test incidents when a discrepancy is observed.



2. Analyze the test incident for further information on the problem
3. Establish the cause of the test incident, e.g. system under test, test documentation, test data or test execution mistake
4. Assign an initial priority and severity level to the test incident
5. Formally report the test incident using an incident classification scheme

*Examples of elements of a test incident report include the following [IEEE 829]:*

- Test incident report identifier
- Summary
- Incident description (input, expected results, actual results, anomalies, date and time, test procedure step, environment, attempts to repeat, testers, observers)
- Priority level
- Severity level

6. Review the test incident report with stakeholders
7. Test incidents are stored in a central repository

### SP 3.4 Write test log

*Test logs are written to provide a chronological record of relevant details about the execution of the tests.*

#### **Typical work products**

1. Test logs

#### **Sub-practices**

1. Collect test execution data
2. Document the test execution data by means of a test log, based on the test log standard

*Examples of elements of a test log include the following [IEEE 829]:*

- Test log identifier
- Description (items being tested, environment in which the testing has been executed)
- Activity and event entries (execution description, test results, anomalous events, incident report identifiers)

3. Review the test log with stakeholders

## SG 4 Manage Test Incidents To Closure

*Test incidents are managed and resolved as appropriate.*

### SP 4.1 Decide on incidents in configuration control board

*Appropriate actions on test incidents are decided upon by a configuration control board (CCB).*

#### **Typical work products**

1. CCB meeting report, including a decision log regarding test incidents
2. Updated incident report

#### **Sub-practices**

1. Establish a CCB with participation of stakeholders, including testing
2. Review and analyze the incidents found
3. Revisit the priority and severity level of the test incident
4. Decide on the test incidents found

*Examples of decisions that can be taken include the following:*

- Rejected; incident is not a defect
- Deferred; incident is declined for repair but may be dealt with a later stage
- Fix; incident is accepted and shall be re-paired

5. Record the decision and other relevant information in the incident database; the incident report is updated.
6. Assign the incident to engineering to perform appropriate actions

**SP 4.2 Perform appropriate action to fix the test incidents**

*Appropriate actions are taken by (test) engineers to fix, re-test and close the test incidents.*

**Typical work products**

1. Test log (including test results)
2. Updated incident report

**Sub-practices**

1. Repair the incident which may involve updating documentation and/or software code.
2. Record information on the repair action in the incident database; the incident report is updated.
3. Perform re-testing, and possibly regression testing, to confirm the fix of the incident.
4. Record information on the re-testing action in the incident database; the incident report is updated.
5. Formally close the incident provided re-testing was successful.

**SP 4.3 Track the status of test incidents**

*The status of the test incidents is tracked and appropriate actions are taken if needed.*

**Typical work products**

1. CCB meeting report
2. Incident status report

**Sub-practices**

1. Provide status reports on incidents to stakeholders.

*Examples of elements that are covered in an incident status report include the following:*

- Incidents opened during period XXXX-XXXX
- Incidents closed during period XXXX-XXXX
- Incidents remaining open for X or more weeks

2. Discuss status reports in a CCB meeting
3. Take appropriate action if needed, e.g. if an incident that needs repair has the same status for a certain period of time.

**Generic Practices by Goals****GG 2 Institutionalize a Managed Process****GP 2.1 Establish an organizational policy**

*Establish and maintain an organizational policy for planning and performing the test design and execution process.*

**Elaboration**

The test design and execution policy typically specifies:

- A set of suitable test design techniques will be identified per test level
- Test specifications will be supported by templates and standards
- Test execution will be done using documented test procedures
- The level of test execution automation required
- Test incidents are documented and reported using an incident classification scheme
- Reported test incidents are evaluated, classified and processed according to a documented procedure
- A basic central test incident repository is put into place

**GP 2.2 Plan the process**

*Establish and maintain the plan for performing the test design and execution process.*

**Elaboration**

Typically, the plan for performing the test design and execution process is included in the test plan, which is described in the TMMi Test Planning process area. The activities for test design and execution are explicitly scheduled as part of the test plan.

**GP 2.3 Provide resources**

*Provide adequate resources for performing the test design and execution process, developing the test work products, and providing the services of the process.*

**Elaboration**

- Adequate time is provided to perform the test design and execution activities
- Experienced individuals, who have expertise in the application domain of the test object and those who have expertise on the development process are available to support the development of the test designs, e.g. participating at reviews
- Tools to support the test design and execution process are available

*Examples of tools include the following:*

- Dynamic analysis tools
- Coverage analysis tools
- Test design tools
- Test data preparation tools
- Capture/playback tools
- Incident management tools

**GP 2.4 Assign responsibilities**

*Assign responsibility and authority for performing the test design and execution process, developing the work products, and providing the services of the test design and execution process.*

**GP 2.5 Train people**

*Train the people performing or supporting the test design and execution process as needed.*

**Elaboration**

Test engineers, and other individuals or groups, involved in test design and execution, are trained in test design and execution and the accompanying procedures and techniques.

*Examples training topics include the following:*

- Formal and informal test design techniques
- Test specification process
- Deriving and prioritizing test conditions and developing test designs
- Development and prioritization of test cases
- Documenting and prioritizing of test procedures
- Test execution activities
- Test specification and test log templates and standards
- Test incident reporting
- Test incident management
- Supporting test design and execution tools

**GP 2.6 Manage configurations**

*Place designated work products of the test design and execution process under appropriate levels of configuration control.*

**Elaboration**

*Examples of work products placed under configuration management include the following:*

- Test design specifications
- Test case specifications

- Test procedure specifications (and/or test scripts)
- Test execution schedule
- Test logs

### GP 2.7 Identify and involve relevant stakeholders

*Identify and involve relevant stakeholders of the test design and execution process as planned.*

#### **Elaboration**

*Examples of activities for stakeholder involvement include:*

- Reviewing and approval of test designs and test cases
- Executing tests, e.g. for validation purposes by end-users
- Participating to the incident management process, e.g. at CCB meetings

### GP 2.8 Monitor and control the process

*Monitor and control the test design and execution process against the plan for performing the process and take appropriate actions.*

#### **Elaboration**

*Examples of measures used to monitor and control the test design and execution process include the following:*

- Number of test specifications completed
- Number of tests executed
- Percentage of tests passed
- Number of outstanding incidents (per priority level)
- Incident trends

### GP 2.9 Objectively evaluate adherence

*Objectively evaluate adherence of the test design and execution process against its process description, standards, and procedures, and address non-compliances.*

#### **Elaboration**

*Examples of review and/or audit evaluation adherence topics include the following:*

- The effectiveness and efficiency of test design techniques
- The compliance of the test specifications (test design, test cases, test procedures) to templates and standards
- The quality of the test cases
- The existence and quality level of the test logs
- Compliance to the incident management process

### GP 2.10 Review status with higher level management

*Review the activities, status and results of the test design and execution process with higher level management and resolve issues.*

## GG 3 Institutionalize a Defined Process

*Only applicable at TMMi level 3*

### GP 3.1 Establish a defined process

*Establish and maintain a description of a defined test design and execution process.*

### GP 3.2 Collect improvement information

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the test design and execution process to support the future use and improvement of the organization's processes and process assets.*

*Examples of measures include the following:*

- Number of test designs established using test design techniques
- Time spent per test specification
- Incident reports by priority and severity
- Effectiveness of test design techniques, e.g. using Defect Detection Percentage (DDP)
- Percentage of test cases automated

## PA 2.5 Test Environment

### Purpose

The purpose of Test Environment is to establish and maintain an adequate environment, including test data, in which it is possible to execute the tests in a manageable and repeatable way.

### Introductory notes

A managed and controlled test environment is indispensable for any testing. It is also needed to obtain test results under conditions which are as close as possible to the 'real-life' situation. This is especially true for higher level testing, e.g. at system and acceptance test level. Furthermore, at any test level the reproducibility of test results should not be endangered by undesired or unknown changes in the test environment.

Specification of test environment requirements is performed early in the projects. The requirements specification is reviewed to ensure their correctness, suitability, feasibility and its representativeness towards the 'real-life' operational environment. Early requirements specification has the advantage that there is more time to acquire and/or develop the required test environment and components such as simulators, stubs or drivers. The type of environment required will depend on the product to be tested and the test types, methods and techniques used.

Availability of a test environment encompasses a number of issues which need to be dealt with: Is it necessary for testing to have an environment per test level? A separate test environment per test team or per test level can be very expensive. Maybe it is possible to have the same environment shared between testers and developers. But then strict management and control is necessary as both testing and development activities are done in the same environment. When poorly managed, this situation can cause many problems ranging from conflicting reservations to people finding the environment in an unknown or undesired state when starting one's activities.

Finally test environment management also includes amongst other managing access to the test environment by providing log-in details, managing test data, configuration management and providing technical support on progress disturbing issues during test execution.

As part of the test environments process area also the requirements regarding generic test data, and the creation and management of the test data is addressed. Whereas specific test data is defined during the test design and analysis activity, more generic test data is often defined and created as a separate activity. Generic test data is re-used by many testers and provides overall background data that is needed to perform the system functions. Generic test data often consists of master data and some initial content for primary data. Sometimes timing requirements influence this activity.

### Scope

The process area Test Environment addresses all activities for specifying test environment requirements, implementing the test environment and managing and controlling the test environment. Managing and control of the test environment also includes aspects such as configuration management and ensuring availability. The test environment process area has both the physical test environment and the test data within its scope.

### Specific Goal and Practice Summary

- SG1 Develop test environment requirements
  - SP 1.1 Elicit test environment needs
  - SP 1.2 Develop the test environment requirements
  - SP 1.3 Analyze the test environment requirements
- SG2 Perform test environment implementation
  - SP 2.1 Implement the test environment
  - SP 2.2 Create generic test data
  - SP 2.3 Specify test environment intake test procedure
  - SP 2.4 Perform test environment intake test
- SG3 Manage and control test environments
  - SP 3.1 Perform systems management
  - SP 3.2 Perform test data management
  - SP 3.3 Co-ordinate the availability and usage of the test environments
  - SP 3.4 Report and manage test environment incidents

## Specific Practices by Goals

### SG 1 Develop Test Environment Requirements

*Stakeholder needs, expectations and constraints are collected and translated into test environment requirements.*

#### SP 1.1 Elicit test environment needs

*Elicit test environment needs, including generic test data, expectations and constraints.*

##### **Typical work products**

1. Test environment needs

##### **Sub-practices**

1. Study the test approach and test plan for test environment implications
2. Engage testing representatives for eliciting test environment needs, including generic test data, expectations and constraints

*Examples of test environment needs include the following:*

- Network components
- Software components, e.g. operating systems, firmware
- Simulator, stubs and drivers
- Supporting documentation, e.g. user guides, technical guides and installation manuals
- Interface components or products
- Tools to develop stubs and drivers
- Test equipment
- Number of test environments
- Generic test databases
- Test data generators
- Test data storage needs
- Test data archive and restore facilities

3. Document the test environment needs, including generic test data, expectations and constraints

#### SP 1.2 Develop the test environment requirements

*Transform test environment needs into test environment requirements*

##### **Typical work products**

1. Test environment requirements
2. Requirements allocation sheet

##### **Sub-practices**

1. Translate the test environment needs, including generic test data, expectations and constraints into documented test environment requirements
2. Allocate test environment requirements to test environment components

#### SP 1.3 Analyze the test environment requirements

*Analyze the requirements to ensure they are necessary, sufficient and realizable.*

##### **Typical work products**

1. Test environment requirements analysis report
2. Test environment requirements review log
3. Test environment project risks

##### **Sub-practices**

1. Analyze test environment requirements to determine whether they fully support the test lifecycle and test approach



*Examples of practices to support the analysis of the test environment requirements:*

- Mapping of test environment requirements to test levels
  - Mapping of test environment requirements to test types
2. Identify key test environment requirements that have a strong influence on cost, schedule or test performance
  3. Identify test environment requirements that can be implemented using existing or modified resources.
  4. Analyze test environment requirements to ensure that they are complete, feasible and realizable
  5. Analyze test environment requirements to ensure that it sufficiently represents the 'real-life' situation, especially for higher test levels
  6. Identify test project risks related to the test environment requirements
  7. Review the test environment requirements specification with stakeholders

## **SG 2** Perform Test Environment Implementation

*The test environment requirements are implemented and made available to be used during the test execution.*

### **SP 2.1** Implement the test environment

*Implement the test environment as specified in the test environment requirements specification and according to the defined plan.*

#### **Typical work products**

1. Operational test environment
2. Unit test results for test environment components

#### **Sub-practices**

1. Implement the test environment as specified and according to the defined plan
2. Adhere to applicable standards and criteria
3. Perform unit testing on test environment components as appropriate
4. Develop supporting documentation, e.g. installation, operation and maintenance documentation
5. Revise the test environment components as necessary

*Examples of when the test environment may need to be revised are when problems surface during implementation that could not be foreseen during requirements specification.*

### **SP 2.2** Create generic test data

*Generic test data as specified in requirements specification is created.*

#### **Typical work products**

1. Test data

#### **Sub-practices**

1. Create generic test data required to allow the execution of the tests
2. Anonymize sensitive data in line with the policy when 'real-life' data is used as a source
3. Archive the set of generic test data

### **SP 2.3** Specify test environment intake test procedure

*The test environment intake test (confidence test) to decide whether the test environment is ready to be used for testing is specified.*

#### **Typical work products**

1. Test environment intake checklist
2. Test environment intake test procedure specification
3. Test environment intake test procedure specification review log

#### **Sub-practices**

1. Define a list of checks to be carried out during the intake test of the test environment

2. Develop the test environment intake test procedure based on the checks identified by putting the checks (test cases) in a executable order and including any other information needed for performing the test environment intake test
3. Document the test environment intake test procedure in a test procedure specification, based in the test procedure specification standard
4. Review the test environment intake test procedure specification with stakeholders

Note that this practice is highly related to the practice SP2.3 “Specify intake test procedure” from the process area Test design and Execution and can possibly be combined.

#### **SP 2.4 Perform test environment intake test**

*Perform the test environment intake test (confidence test) to decide whether the test environment is ready to be used for testing.*

##### **Typical work products**

1. Test environment intake test log
2. Incident reports

##### **Sub-practices**

1. Perform the intake test (confidence test) using the documented intake test procedure to decide if the test environment is ready to be used for testing.
2. Document the results of the test environment intake test by means of a test log, based on the test log standard
3. Log incidents when a discrepancy is observed

Note that this practice is highly related to the practice SP3.1 “Perform intake test” from the process area Test Design and Execution and the intake test on the test object and test environment can possibly be combined. Refer to SP 3.3 “Report test incidents” from the process area Test Design and Execution for more information on incident logging.

### **SG 3 Manage And Control Test Environments**

*Test environments and managed and controlled to allow for uninterrupted test execution.*

#### **SP 3.1 Perform systems management**

*System management is performed on the test environments to effectively and efficiently support the test execution process.*

##### **Typical work products**

1. System management log file
2. Test logging

##### **Sub-practices**

1. Install components needed, e.g. for a specific test session
2. Manage access to the test environment by providing log-in details
3. Provide technical support on progress disturbing issues during test execution
4. Provide logging facilities, which can be used in afterwards to analyze test results

#### **SP 3.2 Perform test data management**

*Test data is managed and controlled to effectively and efficiently support the test execution process.*

##### **Typical work products**

1. Archived test data
2. Test data management log file

##### **Sub-practices**

1. Manage security and access to the test data
2. Manage test data, e.g. with respect to storage resources needed
3. Archive and restore test data and other files on a regular basis and if necessary related to a test session

**SP 3.3 Co-ordinate the availability and usage of the test environments**

*The availability and usage of the test environments by multiple groups is coordinated.*

**Typical work products**

1. Test environment reservation schedule

**Sub-practices**

1. Set up a procedure for managing the usage of test environments by multiple groups
2. Make documented reservations of the test environments in the reservation schedule
3. Identifying specific test environments components needed when making a reservation
4. Discuss conflicting reservation with involved groups and stakeholders
5. Define a test environment reservation schedule for the upcoming period
6. Use the test environment when reserved and a time-slot is assigned
7. Shut down the test environment correctly after usage, e.g. by making sure it is in a known state and test files are removed.

**SP 3.4 Report and manage test environment incidents.**

*Problems that occur when using the test environment are formally reported as incidents.*

**Typical work products**

1. Test environment incident reports
2. CCB meeting reports, including a decision log regarding test environment incidents

**Sub-practices**

1. Log the test environment incident at once when a problem is observed
2. Formally report the test environment incident using an incident classification scheme
3. Manage test environment incidents to closure

Refer to Test Design and Execution process area for practices and Sub-practices covering incident reporting and management.

**Generic Practices by Goals****GG 2 Institutionalize a Managed Process****GP 2.1 Establish an organizational policy**

*Establish and maintain an organizational policy for planning and performing the test environment process.*

**Elaboration**

The test environment policy typically specifies:

- Test environment requirements specification shall be done early in the lifecycle
- Higher levels tests will be carried out in a test environment that is as much as possible 'real-life'
- Management and control of test environments is performed according to documented procedures
- Lower test level, e.g. unit and integration testing, shall apply stubs and drivers for testing
- Privacy and security rules regarding the use of real-life data to create test data

**GP 2.2 Plan the process**

*Establish and maintain the plan for performing the test environment process.*

**Elaboration**

Typically, the plan for performing the test environment process is included in the test plan, which is described in the TMMi Test Planning process area. In a project where the test environment is more complex, and therefore often requires more resources, a specific test environment plan may be established. The plan typically describes amongst others the implementation process of the test environment requirements in detail.

**GP 2.3 Provide resources**

*Provide adequate resources for performing the test environment process, developing the work products, and providing the services of the process.*

**Elaboration**

- Experienced individuals, who have expertise and technical knowledge are available to support the test environment requirements specification
- Adequate time and resources are provided to implement, manage and control the test environment
- Adequate time and resources are provided to create, manage and control the test data
- Adequate time and resources are provided to engineers to develop stubs and drivers needed for low level testing

**GP 2.4 Assign responsibilities**

*Assign responsibility and authority for performing the test environment process, developing the work products, and providing the services of the test environment process.*

**Elaboration**

*Examples of test environment responsibilities to be assigned include the following:*

- Specification of the test environment requirements
- Implementation of test environment
- Configuration management of the test environment
- Solving technical problems related to the test environment
- Ensuring that tests are reproducible with respect to the test environment
- Support and consultancy on test environment-related procedures and technical issues
- Ensuring the availability of the test environment
- Supporting projects in defining an approach for test data
- Creation of generic test data
- Managing test data

**GP 2.5 Train people**

*Train the people performing or supporting the test environment process as needed.*

**GP 2.6 Manage configurations**

*Place designated work products of the test environment process under appropriate levels of configuration control.*

**Elaboration**

*Examples of work products placed under configuration management include:*

- Test environment requirements specification
- Test environment plans
- Test environments
- Test data

Note that configuration management on especially test environments and test data is key to any testing and important for amongst other test reproducibility.

**GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the test environment process as planned.*

**Elaboration**

*Examples of activities for stakeholder involvement include:*

- Reviewing of test environment requirements specification
- Providing resources and/or input for the implementation of the test environment, e.g. subcontractors that develop test environment components

**GP 2.8 Monitor and Control the Process**

Monitor and control the environment process against the plan for performing the process and take appropriate actions.

**GP 2.9 Objectively Evaluate Adherence**

Objectively evaluate adherence of the test environment process against its process description, standards, and procedures, and address non-compliances.

**Elaboration**

Examples of review and/or audit evaluation adherence topics include:

- A test environment requirements specification is written early in the project
- The test environment is as much as possible 'real-life', especially for higher test levels
- The availability of the test environment is at an adequate level
- The effectiveness and efficiency of test environment management and control
- The adequacy of the test data for 'real-life' testing

**GP 2.10 Review status with higher level management**

Review the activities, status and results of the test environment process with higher level management and resolve issues.

**GG 3 Institutionalize a Defined Process**

Only applicable at TMMi level 3

**GP 3.1 Establish a defined process**

Establish and maintain a description of a defined test environment process

**GP 3.2 Collect improvement information**

Collect work products, measures, measurement results, and improvement information derived from planning and performing the test environment process to support the future use and improvement of the organization's processes and process assets.

Examples of measures include the following:

- Number of conflicting test environment reservations
- Effort needed for maintenance, repair and updates
- Number of test case failures due to the test environment
- Average down-time of the test environment
- Number of test environment incidents reported
- Percentage of test environments available on time and according to specification

## TMMi Level 3: Defined

At TMMi level 3, testing is no longer a phase that follows coding. It is fully integrated into the development lifecycle and the associated milestones. Test planning is done at an early project stage, e.g. during the requirements phase, by means of a master test plan. The development of a master test plan builds on the test planning skills and commitments acquired at TMMi level 2. The organization's set of standard test processes, which is the basis for maturity level 3, is established and improved over time. A test organization and a specific test training program exist, and testing is perceived as being a profession. Test process improvement is fully institutionalized as part of the test organization. Organizations at this level understand the importance of reviews in quality control; a formal review program is implemented although not yet fully linked to the dynamic testing process. Reviews take place across the lifecycle. Test professionals are involved in reviews on requirements specifications. Whereby the test designs at TMMi level 2 focus mainly on functionality testing, test designs and test techniques are expanded, depending the business objectives, to also include non-functional testing, e.g. on usability and/or reliability.

A critical distinction between TMMi maturity level 2 and 3 is the scope of the standards, process descriptions, and procedures. At maturity level 2 these may be quite different in each specific instance, e.g. on a particular project. At maturity level 3 these are tailored from the organization's set of standard processes to suit a particular project or organizational unit and therefore are more consistent except for the differences allowed by the tailoring guidelines. Another critical distinction is that at maturity level 3, processes are typically described more rigorously than at maturity level 2. As a consequence at maturity level 3, the organization must revisit the maturity level 2 process areas.

The process areas at TMMi level 3 are:

- 3.1 Test Organization
- 3.2 Test Training Program
- 3.3 Test Lifecycle and Integration
- 3.4 Non-Functional Testing
- 3.5 Peer Reviews

## PA 3.1 Test Organization

### Purpose

The purpose of “Test Organization” is to identify and organize a group of highly skilled people that is responsible for testing. The test group also manages improvements to the organization’s test process and test process assets based on a thorough understanding of the current strengths and weaknesses of the organization’s test process and test process assets.

### Introductory notes

Establishing a test organization implies a commitment to better testing and higher-quality software. It also acquires leadership in areas that relate to testing and quality issues. The staff members of such a group are called test specialists. A test organization (group) is the representation of effective relationships between test specialists, test facilities and project related test activities in order to achieve a high standard in structured testing. The synergy between these elements creates a structure that is more than the sum of the parts. Well-defined communication links from the test group to business, development, and quality assurance are established.

It is important for an organization to have an independent test group. The group shall have a formalized position in the organizational hierarchy. The term independence is used, but each organization must develop its own interpretation and implementation of independence. A test organization can for instance be organized as a test competence centre with a test resource pool whereby group members are assigned to projects throughout the organizations where they do their testing work, or as an independent test group that performs acceptance testing before release. In the TMMi sense, independence for the test organization means that testers are recognized as engineering specialists. Testers are not considered to be developers, and most importantly they report to management independent of development. Test specialists are to be objective and impartial, and unhindered by development organization pressures.

Testing is regarded as a profession and the test group is recognized as a necessity. Detailed and specialized knowledge and skills regarding test engineering, test management and the application domain are characteristics of the motivated individuals assigned to the test group. Test functions and test career paths are defined and supported by a test training program. The group is staffed by people who have the skills and motivation to be good testers. They are assigned to a specific test function and are dedicated to establishing awareness of, and achieving, product quality goals. They measure quality characteristics, and have responsibilities for ensuring the system meets the customers’ requirements. Also the test activities, roles and responsibilities for other staff members (non-test specialists) are specified. For each test function the typical tasks, responsibilities, authorities, required knowledge, skills and test training is specified. As a result the process areas “Test Organization” and “Test Training Program” are closely related and interdependent. One of the principal objectives of the training program is to support the test organization in training of test specialists.

Whereas at TMMi level 2 test process improvement is sometimes an ad-hoc project, it is now well organized and structured within the test organization. The responsibility for facilitating and managing the test process improvement activities, including coordinating the participation of other disciplines, is typically assigned to a test technology manager supported by a management steering committee. Candidate improvements to the process are obtained from various sources, including measurements, lessons learned and assessment results. Careful planning is required to ensure that test process improvement efforts across the organization are adequately managed and implemented. The planning for test process improvement results in a process improvement plan. This plan will address assessment planning, process action planning, pilot planning and deployment planning. When the test improvement is to be deployed, the deployment plan is used. This plan describes when and how the improvement will be implemented across the organization.

### Scope

The process area “Test Organization” defines the functioning (tasks, responsibility, reporting structure) and the position of a test group in the overall organization. Test roles, functions, and career paths are defined to support the acceptance of testing as a professional discipline. Within the test organization, test process improvement is a key activity. Test process improvement encompasses assessing the current test process and using lessons learned to identify possible test improvements, implementing improvement and deploying them in projects.

### Specific Goal and Practice Summary

- SG1 Establish a test organization
  - SP 1.1 Define the test organization
  - SP 1.2 Obtain commitments for the test organization



	SP 1.3	Implement the test organization
SG2	Establish test functions for test specialists	
	SP 2.1	Identify test functions
	SP 2.2	Develop job descriptions
	SP 2.3	Assign staff members to test functions
SG3	Establish test career paths	
	SP 3.1	Establish test career paths
	SP 3.2	Develop personal test career development plans
SG4	Determine, plan and implement test process improvements	
	SP 4.1	Assess the organization's test process
	SP 4.2	Identify the organization's test process improvements
	SP 4.3	Plan test process improvements
	SP 4.4	Implement test process improvements
SG5	Deploy organizational test processes and incorporate lessons learned	
	SP 5.1	Deploy standard test process and test process assets
	SP 5.2	Monitor implementation
	SP 5.3	Incorporate lessons learned into organizational test processes

## Specific Practices by Goals

### SG 1 Establish a Test Organization

*A test organization, which supports the testing practices in projects and the organization, is defined and established.*

#### SP 1.1 Define the test organization

*A test organization is defined and agreed by the stakeholders.*

##### **Typical work products**

1. Test organization description

##### **Sub-practices**

1. Define the test organization, e.g. based on the defined business goals and policy, test goals and policy, and test strategy

*Examples of topics to be addressed when defining a test organization typically include the following:*

- Formal position in the overall organization
- Organizational type
- Level of independence in relation to development
- Tasks, competences and responsibilities of the test organization
- Reporting structure
- Starting points regarding resources, e.g. number of test specialists

Note, that ideally the test organization should be a separate organizational entity or function. However, this is not always possible or practical given the size of the organization, risk level of the systems being developed and the resources available.

2. Review the test organization description with stakeholders

#### SP 1.2 Obtain commitments for the test organization

*Commitments for implementing the test organization are established and maintained.*

##### **Typical work products**

1. Documented requests for commitments
2. Documented commitments

##### **Sub-practices**

1. Identify needed support and negotiate commitments regarding the test organization with relevant stakeholders

2. Document all organizational commitments, both full and provisional
3. Review internal commitments with senior management as appropriate
4. Review external commitments with senior management as appropriate

### SP 1.3 Implement the test organization

*The test organization is implemented in the organization, based on the committed test organization definition.*

#### **Typical work products**

1. Status and results of implementing the test organization

#### **Sub-practices**

1. Track implementation progress and commitments
2. Identify, document, and track to closure issues in implementing the test organization
3. Ensure that the results of implementing the test organization satisfy the organizational goals and objectives

## SG 2 Establish Test Functions For Test Specialists

*Test functions with accompanying job descriptions are established and assigned to the test specialists.*

### SP 2.1 Identify test functions

*A set of test functions is identified, as appropriate.*

#### **Typical work products**

1. List of identified test functions

#### **Sub-practices**

1. Analyze the test policy, test strategy and standard test process for typical test roles
2. Identify a set of test functions that cover the typical test roles, as appropriate

*Examples of test functions include the following:*

- Test manager
- Test team leader
- Test designer
- Test engineer
- Test consultant
- Test environment engineer

3. Identify test functions for specialized areas, as appropriate

*Examples of test functions include the following:*

- Test automation designer
- Test automation engineer
- Performance test engineer
- Usability test engineer
- Test process improvement officer

### SP 2.2 Develop job descriptions

*For the test functions identified, job descriptions are developed. For non test-specialist functions, existing job descriptions are enhanced with typical test tasks and responsibilities.*

#### **Typical work products**

1. Job description for test functions
2. Enhanced job description for non test-specialists

#### **Sub-practices**

1. Define job description for each of the identified test functions

*Job descriptions typically include the following:*

- Name of the test function
- Short description
- Salary scale
- Qualifications
- Typical tasks to be performed
- Responsibilities and authorities
- Required knowledge and skills
- Training modules to be followed

2. Incorporate the job descriptions into the organization's Human Resource Management (HRM) framework
3. Extend job descriptions for other job categories (non-test specialist) to include the test tasks and responsibilities, as appropriate

*Examples of non-test specialists job categories that typically encompass test activities and responsibilities include the following:*

- Software developer
- System engineer
- System integrator
- User representative

4. Use the organization's standard test process as a major input to define and enhance the job descriptions
5. Review the job descriptions with stakeholders

### **SP 2.3 Assign staff members to test functions**

*Members of the test organization are assigned to the identified test functions.*

#### **Typical work products**

1. Staff members assigned to test functions as job title

#### **Sub-practices**

1. Assign staff members to the test functions
2. Perform job interviews to fill open test specialist positions, using questionnaires to determine the interviewee's technical background, his or her personal skills and motivation
3. Ensure that the test specialist positions (functions) are kept occupied
4. Periodically evaluate test organization member performance
5. Take appropriate action based on the evaluation, if necessary

### **SG 3 Establish Test Career Paths**

*Test career paths are established that allow testers' to improve their knowledge, skills, status and rewards.*

#### **SP 3.1 Define test career paths**

*Test career paths are defined that will allow testers to advance their career.*

#### **Typical work products**

1. Test career path framework

#### **Sub-practices**

1. Differentiate within the test functions by creating a junior, intermediate and senior role and provide job descriptions for each of them
2. Link required knowledge and skills, typical tasks and responsibilities, training module and experience level to the junior, intermediate and senior test role for each of the differentiated functions
3. Develop job description for each of the identified differentiated test functions
4. Position the defined and differentiated test functions in a (hierarchical) test career path framework

5. Define a typical time frame that states when one can progress to a next test career path step
6. Link the test career path framework to other career path frameworks available in the organization, e.g. how one can move from being a test manager to a project manager
7. Incorporate the test career framework into the organization's Human Resource Management (HRM) framework

### SP 3.2 Develop personal test career development plans

*A personal test career development plan is developed and maintained for every member of the test organization.*

#### **Typical work products**

1. Personal career development plans

#### **Sub-practices**

1. Personal development plans are developed based on the test career path framework
2. The personal development plan is discussed with the test staff member on a periodic basis
3. Actions are identified and documented that are needed to advance the career development of the staff member
4. Track the defined test career development actions to closure
5. Revise the personal development plan, as appropriate

### SG 4 Determine, Plan and Implement Test Process Improvements

*Strengths, weaknesses, and improvement opportunities for the organization's test process are identified periodically as needed and process actions that address the improvements are planned and implemented.*

#### SP 4.1 Assess the organization's test process

*Assess the organization's test process periodically and as needed to maintain an understanding of its strengths and weaknesses.*

#### **Typical work products**

1. Test process assessment report

#### **Sub-practices**

1. Understand the organization's test process needs using the business goals and policy, test goals and policy, and test strategy
2. Obtain sponsorship for the test process assessments from senior management
3. Define the scope of the test process assessment
4. Plan, schedule and prepare for the test process assessment
5. Conduct the test process assessment
6. Document and present the test assessment report

#### SP 4.2 Identify the organization's test process improvements

*Identify improvements to the organization's test processes and test process assets.*

#### **Typical work products**

1. Prioritized list of test improvements

#### **Sub-practices**

1. Determine candidate test process improvements from test assessment report
2. Prioritize the candidate test process improvements

*Examples of factors that may be helpful to determine the priority of the candidate test process improvements include the following:*

- In synchronization with business and test goals
- According to the maturity model
- Most visible process improvements first to create awareness and acceptance

- Provide measurable and clear business benefits
  - Estimated cost and effort involved
  - Level of difficulty
  - Degree of acceptance
  - Risks mitigated
3. Discuss and review the prioritized list with key stakeholders
  4. Identify and document the test process improvements that will be implemented
  5. Revise the list of planned test process improvements to keep it current

### SP 4.3 Plan test process improvements

*Process actions that address improvements to the organization's test process and test process assets are planned.*

#### **Typical work products**

1. Test process improvement plan

#### **Sub-practices**

1. Identify strategies, approaches, and actions to address the identified test process improvements, e.g. new, unproven, and major changes are piloted before they are incorporated into normal use
2. Establish process action teams to define and implement the test process improvements
3. Document the test process improvement plan

*Examples of elements of a test process improvement plan include the following:*

- Test process improvement objectives
  - Test process improvement organization structure
  - Test process improvements that will be implemented
  - Procedures for monitoring and control
  - Strategies for piloting and implementing test process improvements
  - Responsibilities and authorities
  - Resources and schedules
  - Measurement for determining the effectiveness of the test process improvements
  - Risk associated with the test process improvement plan
4. Review and negotiate the test process improvement plan with stakeholders (incl. members of process action teams)
  5. Review and update the test process improvement plan as necessary

### SP 4.4 Implement test process improvements

*The test process improvements addressed by the test improvement plan are implemented.*

#### **Typical work products**

1. Status and results of implementing test process improvements
2. Plans for test process improvement pilots

#### **Sub-practices**

1. Track progress and commitments against test process improvement plan
2. Plan and run pilots as needed to test selected test process improvements
3. Evaluate results of pilots against plan and with stakeholders
4. Review the activities and work products of process action teams
5. Identify, document, and track to closure issues in implementing the test improvement plan
6. Ensure that the results of implementing test process improvements satisfy the test process improvement objectives

## **SG 5** Deploy Organizational Test Processes and Incorporate Lessons Learned

*The organizational standard test process and test process assets are deployed across the organization and test process-related experiences are incorporated into the organizational test process and test process assets.*

The specific practices within this specific goal describe ongoing activities. Deployment of the standard test process and other organizational test process assets must be continually supported within the organization, particularly for new projects at startup.

### **SP 5.1** Deploy standard test process and test process assets

*Deploy the standard test process and test process assets across the organization, especially to projects at their startup and deploy changes to them as appropriate through the life of each project.*

It is important that not only those that are or will be executing the test process are involved, but also other organizational functions such as (test) training and quality assurance are involved in the deployment as necessary.

#### **Typical work products**

1. Deployment plan
2. Documentation of changes to the organizational standard test process and test process assets
3. Organization's list of projects and status of test process deployment on each project
4. Deployment guidelines and other supporting material for deployment, e.g. training
5. Records of tailoring the organization's standard test process

#### **Sub-practices**

1. Identify projects within the organization that are starting up
2. Identify active projects that would benefit from implementing the organization's (changes to the) standard test process and test process assets
3. Establish plans to deploy the organization's standard test process and test process assets on the identified projects
4. Document the changes to the organizational standard test process and test process assets to enable communication on the changes
5. Ensure that training is available for those who want to start using the standard test process and test process assets
6. Provide guidance and consultation on the use of the standard test process and test process assets
7. Assist projects in tailoring the organization's standard test process and test process assets to meet project needs
8. Maintain records of tailoring and implementing processes on the identified projects and ensure that results from test process tailoring are incorporated into the plan for process-compliance evaluations (see next specific practice)
9. As the organization's standard test process is updated, identify which project should implement the changes

Refer to the process area 'Test Lifecycle and Integration' for more information about how the deployment of organizational test process assets is supported and enabled by the organization's test process asset library.

### **SP 5.2** Monitor implementation

*Monitor the implementation of the organization's standard test process and use of test process assets on projects.*

#### **Typical work products**

1. Results of monitoring test process implementation on projects
2. Status and results of test process-compliance evaluations
3. Results of reviewing selected test process artifacts created as part of process tailoring and implementation

**Sub-practices**

1. Monitor projects for their use of the organization's test process and test process assets and changes to them
2. Review selected test process artifacts created during a project, e.g. for compliance
3. Review the results of test process-compliance evaluations to determine how well the organization's standard test process and test process assets have been deployed
4. Identify, document, and track to closure issues related to implementing the organization's standard test process

**SP 5.3 Incorporate lessons learned into organizational test processes**

*Incorporate lessons learned derived from planning and performing the process into the organizational standard test process and test process assets.*

**Typical work products**

1. Review results regarding the effectiveness and suitability of the standard test process and related test process assets
2. Lesson learned documents (e.g. test evaluation reports)
3. Test process improvement proposals
4. Records of organizational test process improvement activities

**Sub-practices**

1. Conduct periodic reviews of the effectiveness and suitability of the organization's standard test process and related test process assets relative to the business objectives, test goals, test policy and test strategy
2. Obtain feedback about the use of the organization's standard test process and test process assets
3. Derive lessons learned from defining, piloting, implementing and deploying the organization's standard test process and test process assets
4. Make lessons learned available to the people in the organization as appropriate
5. Analyze the organization's test performance indicators and common set of test measures
6. From the information gathered and analyzed, derive test process improvement proposals and software process improvement proposals
7. Submit software process improvement proposals
8. Manage test process improvement proposals

*Examples of activities for managing test process improvement proposals include the following:*

- Soliciting test process improvement proposals
  - Collecting test process improvement proposals
  - Reviewing test process improvement proposals
  - Selecting test process improvement proposals that will be implemented
  - Tracking the implementation of test process improvement proposals
9. Establish and maintain records of the organization's test process improvement activities

**Generic Practices by Goals****GG 2 Institutionalize a Managed Process****GP 2.1 Establish an organizational policy**

*Establish and maintain an organizational policy for planning and performing the test organization process.*

**Elaboration**

The test organization policy typically specifies:

- The test group is recognized as an organizational entity
- Tasks and responsibilities, and the position of the test group in the overall organization
- The level of independence of the test group within the overall organization and projects



- Testing is recognized as a profession
- Test functions and career paths are identified and institutionalized

*Examples of test functions include:*

- Test manager
  - Test team leader
  - Test designer
  - Test engineer
  - Test consultant
  - Test environment engineer
- The standard test process (including templates) is defined and maintained by the test organization, and consistently applied
  - The approach to test metrics, test databases, test tools, and test re-use
  - The test organization facilitates and/or co-ordinates the test activities carried out in projects
  - Each (test) project will provide a test evaluation report (lessons learned) to be used for improving the standard test process
  - The objectives and organizational structure regarding test process improvement
  - The approach for planning, implementing and deploying test process improvements across the organization

## **GP 2.2 Plan the process**

*Establish and maintain the plan for performing the test organization process.*

### **Elaboration**

The plan called for in this generic practice addresses the comprehensive planning for all of the specific practices in this process area required to achieve the specific goals. The “test process improvement plan” is part of the specific practices within this process area and is therefore not the plan referred to by this generic practice.

## **GP 2.3 Provide resources**

*Provide adequate resources for performing test organization process, developing the test work products, and providing the services of the process.*

### **Elaboration**

- An annual budget is available for test organizational activities, e.g. for test process improvement
- Appropriate facilities and tools are made available to perform the test organizational activities
- Fully operational office environment for the test organization is available

## **GP 2.4 Assign responsibilities**

*Assign responsibility and authority for performing the test organization process, developing the work products, and providing the services of the test organization process.*

### **Elaboration**

A resource manager is designated to be responsible for managing the test group. Managing the standard test process is often delegated to a test technology manager. In addition a management steering committee for test process improvement is established and assigned responsibility to provide management sponsorship.

*Examples of test organization responsibilities to be assigned include the following:*

- Representing the test group in the overall organization
- Human resource management for the test specialists and their career paths
- Test process management and improvement
- Facilitating the testing activities carried out by the projects

## **GP 2.5 Train people**

*Train the people performing or supporting the test organization process as needed.*

**Elaboration**

*Examples of training topics include the following:*

- Human resource management training
- Staff appraisal sessions
- Coaching test professionals
- TMMi and other test process improvement reference models
- Planning and managing test process improvement
- Change management

Note that training for (test) engineers and (test) managers on the standard test process and supporting test tools is addressed as part of the process area “Test Training Program”.

**GP 2.6 Manage configurations**

*Place designated work products of the test organization process under appropriate levels of configuration control.*

**Elaboration**

*Examples of work products placed under configuration management include the following:*

- Test organization description
- Job descriptions for test functions
- Test career paths description
- Personal career development plan
- Test assessment reports
- Test process improvement plan
- Deployment plan

**GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the test organization process as planned.*

**Elaboration**

*Examples of stakeholder involvement include the following:*

- Senior management for addressing commitment to a test organization
- Human Resource Management for support and alignment regarding test function descriptions and career development plans
- Process improvement officer for alignment to other process improvement initiatives, e.g. software process improvement

**GP 2.8 Monitor and control the process**

*Monitor and Control the test organization process against the plan for performing the process and take appropriate actions.*

**Elaboration**

*Examples of measures used in monitoring and control the test organization process include the following:*

- Actual number of test specialists per test function vs. planned number of test specialists per test function
- Percentage of test specialists for which a personal test career development plan exists
- Number of test process improvement proposals submitted, accepted or implemented
- A deployment schedule for an organization test process asset
- Percentage of projects using the current organization’s set of standard test processes (or tailored version of the same)

**GP 2.9 Objectively evaluate adherence**

*Objectively evaluate adherence of the test organizational process against its process description, standards, and procedures, and address non-compliances.*

**Elaboration**

*Examples of review and/or audit evaluation adherence topics include the following:*

- Operational performance test organization
- Test staff members assigned to defined test function
- Career development plans
- Planning and coordinating test process improvement activities
- Deploying the organization's set of standard test processes on projects
- Test process improvement plans
- Test process deployment plans

**GP 2.10 Review status with higher level management**

*Review the activities, status and results of the test organization process with higher level management and resolve issues.*

**Elaboration**

*Examples of issues to be reviewed with higher level management include the following:*

- Performance of test organization
- Number of open test positions
- Status of improvements being developed by action teams
- Results of pilots and deployments

**GG 3 Institutionalize a Defined Process****GP 3.1 Establish a defined process**

*Establish and maintain a description of a defined test organization process*

**GP 3.2 Collect improvement information**

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the test organization process to support the future use and improvement of the organization's processes and process assets.*

**Elaboration**

*Examples of measures include the following:*

- Number of test specialists in test organization
- Test employee turnover
- Level of application regarding the standard test process
- Assessment findings that address strengths and weaknesses of the organization's test process
- Status of improvement activities against schedule

## PA 3.2 Test Training Program

### Purpose

The purpose of Test Training Program is to develop a training program, which facilitates the development of knowledge and skills of people so that test tasks and roles can be performed effectively and efficiently.

### Introductory notes

Test Training Program includes training to support the organization's strategic business objectives and to meet the training needs that are common across projects. Specific training needs identified by individual projects are handled at project level. Test Training Program is closely related to and interdependent with the Test Organization process area. One of the main objectives of the Test Training Program is to support the test organization by training the test specialists and other stakeholders involved. A quality training program ensures that those involved in testing continue to improve their testing skills and update their domain knowledge and other knowledge related to testing. The training program may be organized and managed by means of a dedicated training group.

Establishing a Test Training Program is an additional commitment by management to support high-quality testing staff and to promote continuous test process improvement. In testing, a variety of skills is needed. The main categories are test principles, test techniques, test management, test tools, domain knowledge, IT knowledge, system engineering, software development and interpersonal skills. A test training program, consisting of several training modules, is developed to address these categories. Note at higher levels of TMMi other more advanced training categories will become important, e.g. statistical testing at TMMi level 5. Some skills are effectively and efficiently imparted through informal vehicles (e.g. training-on-the-job and mentoring) whereas other skills need formal training to be effectively and efficiently imparted.

The term "training" is used throughout this process area to include all of these learning options. The test training program is linked to the test functions and test roles, and will facilitate test career paths. Deploying the training program guarantees the appropriate knowledge and skill level for all people involved in testing. The implementation of the test training program involves first identifying the organizational test training needs, developing or acquiring specific training modules, subsequently conducting training to address the identified needs as required and finally evaluating the effectiveness of the training program.

### Scope

The process area Test Training Program addresses the establishment of an organizational test training plan and test training capability. It also addresses the actual delivery of the planned test training. Project specific training needs are not part of this process area. They are addressed in the process area Test Planning.

### Specific Goal and Practice Summary

- SG1 Establish an organizational test training capability
  - SP 1.1 Establish the strategic test training needs
  - SP 1.2 Align the organizational and project test training needs
  - SP 1.3 Establish an organizational test training plan
  - SP 1.4 Establish test training capability
- SG2 Provide necessary test training
  - SP 2.1 Deliver test training
  - SP 2.2 Establish test training records
  - SP 2.3 Assess test training effectiveness

### Specific Practices by Goals

#### SG 1 Establish an Organizational Test Training Capability

*A training capability, which supports the organizations test roles, is established and maintained.*

#### SP 1.1 Establish the strategic test training needs

*Establish and maintain the strategic test training needs of the organization.*

##### **Typical work products**

1. Training needs

## 2. Assessment analysis

### Sub-practices

1. Analyze the organizations strategic business objectives, test policy and strategy and (test) process improvement plan to identify current and potential future test training needs

*Examples of categories of test training needs include the following:*

- Test engineering and process (e.g. organizational standard test process, test principles, test lifecycle, static test techniques, dynamic test techniques, test tools and test automation)
- Test management (e.g. test estimation, tracking, and risk management)
- IT related training (e.g. requirements engineering, configuration management, project management, system engineering, software development, development lifecycle models)
- Interpersonal skills (e.g. communication, team building)
- Domain expertise

Note the identification of test process training is primarily based on the skills that are required to perform the organization's set of standard test processes.

2. Periodically assess the test skill set of the people involved in testing
3. Document the strategic test training needs of the organization
4. Map the test training needs to the test functions (incl. test career paths) and test roles of the organization
5. Revise the organizations strategic test training needs as necessary

### SP 1.2 Align the organizational and project test training needs

*Align the organizational and project test training needs and determine which test training needs are the responsibilities of the organization and which will be left to the individual projects.*

The organization's training staff is responsible for addressing common cross-project test training needs. In some cases, however, the organization's training staff may address additional test training needs of projects within the context of the training resources and the organization's training priorities.

#### Typical work products

1. Common project test training needs
2. Training commitments to projects

### Sub-practices

1. Analyze the test training needs identified by various projects

Analysis of specific project needs is intended to identify common test training needs that can be most efficiently addressed organization-wide. This analysis activity can also be used to anticipate future test training needs that are first visible at the project level.

2. Determine whether the training needs identified by the various projects are project specific or common to the organization

Test training needs common to the organization are normally managed by means of the organizational test training program.

3. Negotiate with the various projects on how their specific training needs will be satisfied

*Examples of training appropriately performed by the project include the following:*

- Training in the application domain of the project
- Training in the unique tools and methods used by the project

4. Document the commitments for providing test training support to the projects

Refer to SP 4.2 "Plan for test staffing" from the process area "Test Planning" for more information on project specific plan for training.

### SP 1.3 Establish an organizational test training plan

*Establish and maintain an organization tactical test training plan.*

Note that in many organizations this planning is performed annually with a review each quarter.

**Typical work products**

1. Test training plan
2. Test training commitments

**Sub-practices**

1. Establish test training plan content

*Examples of elements of an organizational tactical test training plan include the following:*

- Test training topics
  - Schedules based on test training activities and their dependencies
  - Methods used for training
  - Requirements and quality standards for training materials
  - Training tasks, roles and responsibilities
  - Required resources including tools, facilities, environments, staffing, and skills and knowledge
  - Data to be collected for measuring training effectiveness
2. Review test training plan with affected groups and individuals, e.g. human resources, test resources and project management.
  3. Establish commitment to the test training plan
  4. Revise test training plan and commitments as necessary

**SP 1.4 Establish test training capability**

*Establish and maintain test training capability to address the organizational training needs.*

**Typical work products**

1. Test training materials and supporting artifacts

**Sub-practices**

1. Select the appropriate approaches to satisfy specific test training needs

*Examples of training approaches include the following:*

- Classroom training
  - Computer-aided instruction
  - Guided self-study
  - Formal apprenticeship and mentoring programs
  - Brown-bag lunch seminars
  - Structured on-the-job training
2. Determine whether to develop test training materials internally or acquire them externally
- Example criteria that can be used to determine the most effective mode of knowledge or skill acquisition include the following:*
- Time available to prepare for project execution
  - Availability of in-house expertise
  - Availability of training from external sources
  - Available budget
  - Time required for maintenance of training material
3. Develop or obtain test training materials
  4. Develop or obtain qualified instructors
  5. Describe the training in the organization's test training curriculum

*Examples of the information provided in the test training descriptions for each course include the following:*

- Training objectives
- Topics covered in the training
- Intended audience
- Prerequisites, e.g. other training courses and practical experience

- Preparation for participating
- Length of the training
- Lesson plans
- Completion criteria for the course

6. Revise the test training materials and supporting artifacts as appropriate

*Examples of situations in which the test training materials and supporting artifacts may need to be revised include the following:*

- Test training needs change (e.g., when new technology associated with the training topic is available)
- When evaluation of the test training identifies the need for change (e.g., evaluations of training effectiveness surveys, training program performance assessments, or instructor evaluation forms)

## SG 2 Provide Necessary Test Training

*Training necessary for testers and other individuals involved in testing to perform their role effectively is provided.*

In selecting people to be trained, also consider the need for managers to understand the basic testing principles and test strategy, developers to be able to perform unit and integration testing, users to be able to participate in acceptance testing, etc.

### SP 2.1 Deliver test training

*Deliver the training according to the organizational test training plan.*

#### **Typical work products**

1. Delivered training course
2. Completed course evaluation forms

#### **Sub-practices**

1. Select the people who will receive the training necessary to perform their test role effectively  
Note a waiver may be provided to those that already possess the knowledge and skills required to perform well in their designated roles. Care should be taken that training waivers are not abused.
2. Schedule the training including any required resources, as necessary (e.g. facilities and instructors)
3. Conduct the training
4. Complete course evaluation form by participants
5. Track the delivery of training against the plan

### SP 2.2 Establish test training records

*Establish and maintain records of the organizational test training conducted.*

Although strictly speaking the scope of this process area is for test training performed at the organizational level, to provide consistent and complete information on each employee, the training records preferably include all training, whether performed at organizational level or at project level.

#### **Typical work products**

1. Test training records
2. Training updates to the organizational repository

#### **Sub-practices**

1. Keep records for all employees that successfully complete a training course or other training activity as well as those who have been unsuccessful
2. Keep records of all employees who have been waived from specific training including rationale and management approval
3. Make training records available to the appropriate people for consideration in assignments, e.g. by providing a skill matrix with a summary of experience and education of people



**SP 2.3 Assess test training effectiveness**

*Assess the effectiveness of the organization's test training program.*

The results of the test training effectiveness assessments should be used to revise training materials as described in the "Establish training capability" specific practice.

**Typical work products**

1. Training effectiveness surveys
2. Training program performance assessments
3. Training examinations results

**Sub-practices**

1. Assess in-progress or completed projects to determine whether employee knowledge is adequate for performing project test tasks
2. Assess the effectiveness of each training course with respect to established organizational, project, or individual learning objectives
3. Obtain student evaluations of how well training activities met their needs

**Generic Practices by Goals****GG 2 Institutionalize a Managed Process****GP 2.1 Establish an organizational policy**

*Establish and maintain an organizational policy for planning and performing the organizational test training process.*

**Elaboration**

The test training policy typically specifies:

- The knowledge and skills needed for performing the test functions and roles are identified
- Test training vehicles for imparting knowledge and skills are identified
- Test training is provided to build a knowledge and skill base for testing, to fulfill the needs of projects and to develop the skills of the individuals
- An in-house training group is established
- Test training is developed within the organization or obtained from outside the organization when appropriate
- Test training is also applicable for business representatives, software engineers, integrators and architects that fulfill a test role within a project

**GP 2.2 Plan the process**

*Establish and maintain the plan for performing the organizational test training process.*

**Elaboration**

This plan for performing the test training process differs from the test training plan described in a specific practice in this process area. The plan for this generic practice would address the comprehensive planning for all of the specific practices in this process area, from the establishment of strategic test training needs all the way through the assessment of the effectiveness of the test training effort. In contrast the test training plan would address the periodic planning for the delivery of individual training offerings.

**GP 2.3 Provide resources**

*Provide adequate resources for performing the organizational test training process, developing the test work products, and providing the services of the process.*

**Elaboration**

- An annual budget is available for test training
- People, e.g. the organizational training staff, with the appropriate skills are available

*Examples of people (full or part time, internal or external), and skills include the following:*

- Testing experts
  - Domain experts
  - Curriculum designers
  - Course designers
  - Instructors
  - Training administrators
- Appropriate facilities and tools are made available to perform training

*Examples of training facilities and tools include the following:*

- Classroom training facilities
- Workstations to be used for training
- Computer based training packages
- Packages for developing presentation materials

#### **GP 2.4 Assign responsibilities**

*Assign responsibility and authority for performing the organizational test training process, developing the work products, and providing the services of the test training process.*

##### **Elaboration**

A group (or person) is designated to be responsible for developing, managing and co-coordinating the test training program, e.g. organizational training department/coordinator, human resources, etc.

#### **GP 2.5 Train people**

*Train the people performing or supporting the organizational test training process as needed.*

##### **Elaboration**

*Examples of training topics include the following:*

- Knowledge and skill needs analysis
- Course design
- Training delivery techniques/methods
- Refreshing training on subject matter

#### **GP 2.6 Manage configurations**

*Place designated work products of the organizational test training process under appropriate levels of configuration control.*

##### **Elaboration**

*Examples of work products placed under configuration management include the following:*

- Test training plan
- Training records
- Training materials and supporting artifacts
- Evaluation forms

#### **GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the organizational test training process as planned.*

##### **Elaboration**

*Examples of activities for stakeholder involvement include the following:*

- Identifying test training needs
- Reviewing the test training plan
- Assessing test training effectiveness

#### **GP 2.8 Monitor and control the process**

*Monitor and Control the organizational test training process against the plan for performing the process and take appropriate actions.*

**Elaboration**

*Examples of measures used in monitoring and control the test training process include the following:*

- Number of training courses delivered (e.g. planned versus actual)
- Actual attendance at each training course compared to the projected attendance
- Schedule for delivery of training
- Schedule for development of courses
- Training costs against allocated budget
- Progress in developing and providing training courses compared to the documented test training needs

**GP 2.9 Objectively evaluate adherence**

*Objectively evaluate adherence of the organizational test training process against its process description, standards, and procedures, and address non-compliances.*

**Elaboration**

*Examples of review and/or audit evaluation adherence topics include the following:*

- Process for developing and revising the training plan
- Process for developing and revising training courses
- Providing necessary test training
- Test training plan
- Test training records
- Training materials and supporting artifacts
- Instructor evaluation forms

**GP 2.10 Review status with higher level management**

*Review the activities, status and results of the organizational test training process with higher level management and resolve issues.*

**Elaboration**

*Examples of issues to be reviewed with higher level management include the following:*

- The effectiveness of the test training program
- Progress regarding test training activities
- Test training costs
- The performance of subcontracted training organizations

**GG 3 Institutionalize a Defined Process****GP 3.1 Establish a defined process**

*Establish and maintain a description of a defined organizational test training process*

**GP 3.2 Collect improvement information**

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the organizational test training process to support the future use and improvement of the organization's processes and process assets.*

*Examples of measures include the following:*

- Number of training courses delivered (e.g. planned versus actual)
- Post-training evaluation ratings
- Training program quality survey ratings

## PA 3.3 Test Lifecycle and Integration

### Purpose

The purpose of “Test Lifecycle and Integration” is to establish and maintain a usable set of organizational test process assets (e.g. a standard test lifecycle) and work environments standards and to integrate and synchronize the test lifecycle with the development lifecycle. The integrated lifecycle will ensure early involvement of testing in project. The purpose of “Test Lifecycle and Integration” is also to define a coherent test approach across multiple test levels, based on the identified risks and the defined test strategy, and to provide an overall test plan, based on the defined test lifecycle.

### Introductory notes

An important responsibility of the test organization is define, document and maintain a standard test process, in line with the organizations’ test policy and goals. Organizational test process assets enable consistent test process performance across the organization and provide a basis for cumulative, long- term benefits to the organization. The organization’s test process asset library is a collection of items maintained for use by the people and projects of the organization. The collection of items include descriptions of test processes, descriptions of test lifecycle models (including supporting templates and guidelines for the test deliverables), supporting test tools, process tailoring guidelines and a test process database. The organization’s test process asset library supports organizational learning and process improvement by sharing of best practices and lessons learned across the organization. The standard test lifecycle models define the main phases, activities and deliverables for the various test levels. The testing activities will subsequently be performed in project according to these models. Standards and guidelines are developed for test related (work) product. The standard test lifecycle models are aligned with the development lifecycle models to integrate the testing activities in terms of phasing, milestones, deliverables, and activities. Life cycle integration is done in such a way that early involvement of testing in projects is ensured, e.g. test planning starts during the requirements specification phase. Integration and unit test planning are initiated at detailed design time. Testers will review the test basis documents to determine its testability, and development planning can be influenced by considering the test approach. The organization’s set of standard test processes can be tailored by projects to create their defined processes. The work environment standards are used to guide creation of project work environments.

At TMMi level 3, test management is concerned with master test planning which addresses the co-ordination of testing tasks, responsibilities and test approach over multiple test levels. This prevents unnecessary redundancy or omissions of tests between the various test levels and can significantly increase the efficiency and quality of the overall test process. The information resulting from project test planning is documented in a project test plan, which governs the detailed phase test plans to be written specifically for an individual test level. The master test plan describes the application of the test strategy for a particular project, including the particular levels to be carried out and the relationship among those levels. The master test plan should be consistent with the test policy and strategy, and, in specific areas where it is not, should explain those deviations and exceptions. The master test plan will complement the project plan or operations guide which describes the overall test effort as part of the larger project or operation. The Master test plan provides an overall test planning and test management document for multiple levels of test (either within one project or across multiple projects). On smaller projects or operations (e.g. where only one level of testing is formalized) the master test plan and the test plan will often be combined into one document.

### Scope

The process area “Test Lifecycle and Integration” addresses all practices to establish and maintain a usable set of organizational test process assets (e.g. a standard test lifecycle) and work environments standards, and to integrate and synchronize the test lifecycle with the development lifecycle. “Test Lifecycle and Integration” also addresses the master test planning practices. The master test plan at TMMi level 3 defines a coherent test approach across multiple test levels.

### Specific Goal and Practice Summary

- SG1 Establish organizational test process assets
  - SP 1.1 Establish standard test processes
  - SP 1.2 Establish test lifecycle model descriptions addressing all test levels
  - SP 1.3 Establish tailoring criteria and guidelines
  - SP 1.4 Establish the organization’s test process database
  - SP 1.5 Establish the organization’s test process asset library

	SP 1.6	Establish work environment standards
SG2	Integrate the test lifecycle models with the development models	
	SP 2.1	Establish integrated lifecycle models
	SP 2.2	Review integrated lifecycle models
	SP 2.3	Obtain commitments on the role of testing within the integrated lifecycle models
SG3	Establish a master test plan	
	SP 3.1	Perform product risk assessment
	SP 3.2	Establish the test approach
	SP 3.3	Establish test estimates
	SP 3.4	Define the organization
	SP 3.5	Develop the master test plan
	SP 3.6	Obtain commitment to the master test plan

## Specific Practices by Goals

### SG 1 Establish Organizational Test Process Assets

*A set of organizational test process assets is established and maintained.*

#### SP 1.1 Establish standard test processes

*Establish and maintain the organization's set of standard test processes.*

Multiple standard test processes may be needed to address the needs of different application domains, test levels, lifecycle models, methodologies, and tools. The organization's set of standard test processes typically focuses on technical processes. However, as needed management, administrative, support and organizational process can also be part of the standard test process framework. The organization's set of test processes should collectively cover all processes needed by the organization and projects, including those processes addressed at maturity level 2.

##### **Typical work products**

1. Organization's set of standard test processes

##### **Sub-practices**

1. Decompose each standard test process into constituent process elements to the detail needed to understand and describe the process
2. Specify the critical attributes of each process element

*Examples of critical elements include the following:*

- Process roles and responsibilities
- Applicable standards
- Applicable procedures, methods, and tools
- Entry criteria
- Inputs
- Product and process measures to be collected
- Verification points (e.g. reviews)
- Outputs
- Interfaces
- Exit criteria

3. Specify the relationships of the process elements

*Examples of relationships include the following:*

- Sequence of process elements
- Interfaces among the process elements
- Interfaces with process external to testing
- Interdependencies among process elements

4. Ensure that the organization's set of standard test processes adheres to applicable policies, standards, and models

Adherence to applicable standards and models is typically demonstrated by developing a mapping from the organization's set of standard test processes to the relevant standards and models.

5. Ensure the organization's set of standard test processes satisfies the test process needs and objectives of the test organization
6. Document the organization's set of standard test processes
7. Conduct peer reviews on the organization's set of standard test processes
8. Revise the organization's set of standard test processes as necessary

### SP 1.2 Establish test lifecycle model descriptions addressing all test levels

*Establish and maintain descriptions of the test lifecycle models (including supporting templates and guidelines for the test deliverables), approved for use in the organization, thereby addressing all identified test levels.*

#### Typical work products

1. Description of test lifecycle models

#### Sub-practices

1. Select test lifecycle models based on the needs of the projects and the organization
2. Document the descriptions of the test lifecycle models

*A test lifecycle model description typically includes the following:*

- Test strategy, e.g. test levels and their objectives
- Test lifecycle phases, e.g. planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities
- Entry and exit criteria for each phase
- Testing activities per phase
- Responsibilities
- Deliverables
- Milestones

3. Develop supporting templates and guidelines for the deliverables identified within the test lifecycle models

*Examples of test deliverables that are supported by means of templates and guidelines typically include the following:*

- Master test plan
- Phase test plan
- Test design specification
- Test case specification
- Test procedure specification
- Test log
- Test summary report
- Test evaluation report

4. Conduct peer reviews on the test lifecycle models, and supporting templates and guidelines
5. Revise the description of the test lifecycle models, and supporting templates and guidelines, as necessary

### SP 1.3 Establish tailoring criteria and guidelines

*Establish and maintain the tailoring criteria and guidelines for the organization's set of standard test processes.*

#### Typical work products

1. Tailoring criteria and guidelines for the organization's set of standard test processes

*Tailoring criteria and guidelines typically include the following:*

- How the organization's set of standard test processes and organizational test process assets are used to create tailored defined test processes
- Mandatory requirements that must be satisfied by the tailored defined processes
- Options that may be exercised and criteria for selecting among the options
- Procedures that must be followed in performing and documenting test process tailoring

### **Sub-practices**

1. Specify the selection criteria and procedures for tailoring the organization's set of standard test processes

*Examples of tailoring actions include the following:*

- Modifying a test lifecycle model
- Combining elements of different test lifecycle models
- Modifying test process elements
- Replacing test process elements
- Deleting test process elements
- Reordering test process elements

2. Specify the standards for documenting the tailored test processes
3. Specify the procedures for submitting and obtaining approval of waivers from requirements of the organization's set of standard test processes
4. Document the tailoring guidelines for the organization's set of standard test processes
5. Conduct peer reviews on the tailoring guidelines
6. Revise the tailoring guidelines as necessary

## **SP 1.4 Establish the organization's test process database**

*Establish and maintain the organization's test process database.*

### **Typical work products**

1. Definition of the common set of test process and product data for the organization's set of standard test processes
2. Organization's test process database repository (that is, the repository structure and support environment)
3. Organization's test process database

### **Sub-practices**

1. The test process database is established to collect and make available data on the test processes and resulting work products

*Examples of test process and work product data typically include the following:*

- Test estimates and actual data, e.g. on size, effort and cost
- Quality measures, e.g. number of defects found by priority level
- Peer review coverage
- Test coverage
- Reliability measures

2. The data entered into the test process database are reviewed to ensure the integrity of the database content

The test process database also contains or references the actual measurement data and related information and data needed to understand and interpret the measurement data and access it for reasonableness and applicability.

3. The test process database is managed and controlled

User access to the test process database contents is controlled to ensure completeness, integrity, security and accuracy of the data



**SP 1.5 Establish the organization's test process asset library**

*Establish and maintain the organization's test process asset library.*

**Typical work products**

1. Organization's test process asset library
2. Catalog of items in the organization's test process asset library

**Sub-practices**

1. Design and implement the organization's test process asset library, including the library structure and support environment
2. Specify the criteria for including item in the library, e.g. primarily based on their relationship to the organization's set of standard test processes
3. Specify the procedures for storing and retrieving items
4. Enter the selected items into the library and catalog them for easy reference and retrieval

*Examples of items to be stored in the organization's test process asset library typically include the following:*

- Test policy and test strategy
  - Defined test process descriptions
  - Procedures (e.g. test estimation procedure)
  - Templates
  - Best practices test process assets
  - Test plan
  - Training materials
  - Process aids (e.g. checklists)
  - Lesson learned documents (e.g. test evaluation reports)
5. Make the items available for use in projects
  6. Periodically review the use of each item and use the results to maintain the library contents
  7. Revise the organization's test process assets library as necessary

**SP 1.6 Establish work environment standards**

*Establish and maintain work environment standards.*

**Typical work products**

1. Work environment standards

*Examples of work environment standards include the following:*

- Procedures for operation, safety and security of the work environments
- Standard workstation hardware and software
- Standard application software

**Sub-practices**

1. Evaluate commercially-available work environment standards appropriate for the organization
2. Adopt existing work environment standards and develop new ones to fill gaps based on the organization's test process needs and objectives

**SG 2 Integrate the Test Life Cycle with the Development Models**

*A set of organizational test process assets is established and maintained.*

**SP 2.1 Establish integrated lifecycle models**

*Establish and maintain descriptions of the integrated test and development lifecycle models, approved for use in the organization.*

**Typical work products**

1. Description of integrated lifecycle models

**Sub-practices**

1. Synchronize the phases of the test lifecycle models with the phase of the development lifecycle models
2. Ensure testing is involved early in the development lifecycle, e.g. during requirements development
3. Define mutual dependencies with respect to testing and development activities
4. Define mutual dependencies with respect to testing and development deliverables and lifecycle milestones
5. Document the descriptions of the integrated lifecycle models
6. Revise the description of the integrated lifecycle models, as necessary

**SP 2.2 Review integrated lifecycle models**

*Review the integrated lifecycle models to make stakeholders understand the role of testing within the integrated development and test lifecycle.*

**Typical work products**

1. Integrated lifecycle review log

**Sub-practices**

1. Organize reviews with stakeholders to make them understand the role of testing within the integrated development and test lifecycle.

**SP 2.3 Obtain commitments on the role of testing within the integrated lifecycle models**

*Obtain commitments on the role of testing within the integrated lifecycle models from relevant stakeholders responsible for managing, performing and supporting project activities based on the integrated lifecycle models.*

**Typical work products**

1. Documented requests for commitments
2. Documented commitments

**Sub-practices**

1. Identify needed support and negotiate commitments with relevant stakeholders
2. Document all organizational commitments, both full and provisional
3. Review internal commitments with senior management as appropriate
4. Review external commitments with senior management as appropriate

**SG 3 Establish A Master Test Plan**

*A master test plan is established to define a coherent test approach across multiple test levels and an overall test planning.*

**SP 3.1 Perform product risk assessment**

*A product risk assessment is performed to identify the typical critical areas for testing.*

**Typical work products**

1. Product risk list, with a category and priority assigned to each risk

**Sub-practices**

1. Identify and select stakeholders that need to contribute to the product risk assessment
2. Identify generic product risks using input from stakeholders
3. Document the context and potential consequences of the product risk
4. Identify the relevant stakeholders associated for each product risk
5. Review the identified product risks against the test assignment
6. Analyze the identified products risks using the predefined parameters, e.g. likelihood and impact
7. Categorize and group product risks according to the defined risk categories

8. Prioritize the product risks for mitigation
9. Review and obtain agreement with stakeholders on the completeness, category and priority level of the product risks
10. Revise the product risks as appropriate

Refer to SG 1 “Perform a product risk assessment” from the process area “Test Planning” for more details on the (sub) practices for performing the product risk assessment.

### SP 3.2 Establish the test approach

*The test approach is established and agreed upon to mitigate the identified and prioritized product risks.*

#### **Typical work products**

1. Test approach
2. List of items to be tested and not to be tested
3. List of features to be tested and not to be tested
4. Identified set of test levels
5. Allocation table of test items/test features/products risk to test levels
6. Entry criteria per test level
7. Exit criteria per test level

#### **Sub-practices**

1. Identify and document the items and features to be tested, and not to be tested, based on the product risks.

Note that the level of aggregation of the test items and test features is likely to be higher during master test planning than at planning for an individual test level.

2. Identify the test levels that are needed to mitigate the product risks
3. Allocate the items and features to be tested, and product risks to the identified test levels
4. Select the test design techniques to be used at various test levels; multiple test design techniques are defined to vary in test coverage based on the defined product risks
5. Define the approach to review test work products
6. Define the approach for re-testing and regression testing
7. Identify the supporting test tools to be used
8. Identify significant constraints regarding the test approach
9. Define a set of entry criteria related to the test process and to product quality for each identified test level
10. Define a set of exit criteria related to the test process to product quality for each identified test level
11. Align the test approach with the defined organization-wide or program-wide test strategy
12. Identify any non-compliances to the test strategy and its rationale
13. Review the test approach with stakeholders
14. Revise the test approach as appropriate

Refer to SG 2 “Establish a test approach” from the process area “Test Planning” for more details on the (sub) practices for establishing the test approach.

### SP 3.3 Establish test estimates

*Well founded test estimates are established and maintained for use in discussing the test approach and in planning the testing activities.*

Note that often, early in the development lifecycle, not all required information is available to establish a test estimate. As a consequence, the accuracy of the test estimate is limited. It is important for the test manager to make it clear to the stakeholders that the test estimate will have to be detailed, and possibly adapted, later on in the lifecycle.

#### **Typical work products**

1. Work breakdown structure
2. Selected test lifecycle model

3. Test effort estimates
4. Test cost estimates

#### **Sub-practices**

1. Establish a top-level work breakdown structure (WBS) based on the defined test approach to clearly define the scope of the test estimate.
  2. Select a test lifecycle model from the organization's standard set on which to scope the planning effort
  3. Estimate the test effort and cost for the test work products and tasks based on estimation rationale, e.g. test metrics from the test process database
  4. Align the estimated test effort and costs with the overall estimated project effort and costs
- Refer to SG 3 "Establish test estimates" from the process area "Test Planning" for more details on the (sub) practices for establishing test estimates.

### **SP 3.4 Define the organization**

*The organization within testing at the various levels is defined, including the interfaces to other processes, and a clear overview of what is expected from the various parties involved is established.*

The relationship of testing to other processes such as development, project management, quality assurance, and configuration management is determined and described. This includes the lines of communication within the test organization, the authority for resolving issues raised by testing, and the authority for approving test products and processes. This may include a visual representation, e.g. an organizational chart.

#### **Typical work products**

1. Description of organization

#### **Sub-practices**

1. Determine the test roles at various test levels to ensure alignment between the various test levels
2. Define authorities and responsibilities for the various test roles, products and processes
3. Define the organizational structure, e.g. the relationship between the various roles, the identified test levels and other stakeholders within the development process
4. Define the communication structure (e.g. meetings and reports), both internal within testing and to external stakeholders

### **SP 3.5 Develop the master test plan**

*The master test plan is established to define a coherent test approach across multiple test levels and an overall test plan.*

#### **Typical work products**

1. Master test plan

#### **Sub-practices**

1. Establish the master test schedule with predefined stages of manageable size for the identified test levels based on the developed test estimate and selected test lifecycle
2. Align the master test schedule with the overall project schedule
3. Plan for necessary test staffing resources with the required knowledge and skills to perform the testing
4. Plan the involvement of identified stakeholders
5. Identify, analyze and document the project risks associated with testing
6. Establish and maintain the master test plan

*Examples of elements of a master test plan include the following [after IEEE 829]:*

- Test plan identifier
- An overall introduction (scope, references, system overview and test overview)
- Organization, including roles and responsibilities
- Non-compliances to the test strategy and its rationale
- Items to be tested (including risk level) and not to be tested

- Features to be tested (including risk level) and not to be tested
- Identification of test levels and test types
- Test approach (e.g. test design techniques) per test level
- Entry and exit criteria per test level
- Test milestones and work products
- Test lifecycle and tasks
- Environmental needs and requirements (including office environment)
- Staffing and training needs
- Stakeholder involvement
- Test estimate
- Master test schedule
- Test project risks and contingencies

Refer to SG 4 “Develop a test plan” from the process area “Test Planning” for more details on the (sub) practices for developing a master test plan.

Refer to the “Test Environment” process area for more information on environment needs and requirements.

### SP 3.6 Obtain commitment to the master test plan

*Commitments to the master test plan are established and maintained.*

#### **Typical work products**

1. Documented requests for commitments
2. Master test plan review log
3. Revised and renegotiated master test plan, e.g. test budgets, test schedule, product risk list, stakeholders agreements
4. Documented commitments

#### **Sub-practices**

1. Organize reviews with stakeholders to make them understand test commitments
2. Discuss differences between estimates and available resources with stakeholders
3. Reconcile any differences between estimates and available resources
4. Identify needed support and negotiate commitments with relevant stakeholders
5. Document all organizational commitments, both full and provisional
6. Review internal commitments with senior management as appropriate
7. Review external commitments with senior management as appropriate

Refer to SG 5 “Obtain commitment to the test plan” from the process area “Test Planning” for more details on the (sub) practices for obtaining commitment to the master test plan.

## Generic Practices by Goals

### GG 2 Institutionalize a Managed Process

#### GP 2.1 Establish an organizational policy

*Establish and maintain an organizational policy for planning and performing the test lifecycle and integration process.*

#### **Elaboration**

The policy for test lifecycle and integration typically specifies:

- Expectations for establishing and maintaining a set of standard test processes for use in the organization
- Expectations for making organization test process assets available across the organization
- A standard test lifecycle is defined for each test level
- Standards and guidelines are available for test (work) products at each lifecycle phase

- The defined test lifecycle is integrated with the development lifecycle
- Master test planning co-ordinates test activities over multiple test levels
- The test levels that are governed by means of a master test plan
- Master test planning is aligned with the organization wide or program wide test strategy
- Testing is involved at an early stage of development

## GP 2.2 Plan the process

*Establish and maintain the plan for performing the test lifecycle and integration process.*

### **Elaboration**

The plan for performing establishing and maintaining the organization's standard test processes and test process assets can be part of (or referenced by) the organization's test process improvement plan.

Typically, the plan for establishing the master test plan is included in the project plan, which is described in the CMMI "Project Planning" process area.

## GP 2.3 Provide resources

*Provide adequate resources for performing the test lifecycle and integration process, developing the test work products, and providing the services of the process.*

### **Elaboration**

A test technology manager, supported by a test process group, manages the definition of organization's standard test processes. The test process group is typically staffed by a core of test professionals. The test process group is supported by test process owners, a process improvement manager and people with expertise in various testing and other disciplines.

*Examples of other resources provided for defining and maintaining the organization's standard test processes include the following tools:*

- Database management tools
- Process modeling tools
- Web page builders

*Examples of resources for establishing the master test plan include the following:*

- For the master test plan a documented and approved assignment exists for testing typically covering issues and expectation regarding goals and objectives, exit criteria, items and features to be tested and not to be tested, type of testing to be performed, imposed standards, cost, schedule and resource constraints
- Adequate time is provided to test management to perform the master test planning activities
- Experienced individuals, who have expertise in the application domain of the test object and those who have expertise on the development process are available to support the development of the master test plan
- Tools to support the master test planning process are available, e.g. project planning and scheduling tools, estimation tools, risk assessment tools, test management tools and configuration management tools.

## GP 2.4 Assign responsibilities

*Assign responsibility and authority for performing the test lifecycle and integration process, developing the work products, and providing the services of the test lifecycle and integration process.*

### **Elaboration**

A test technology manager, supported by a test process group, is often designated to be responsible for managing the definition of organization's standard test processes. The test process group is typically staffed by a core of test professionals. The test process group is supported by test process owners, a process improvement manager and people with expertise in various testing and other disciplines.

A test manager is typically designated to be responsible for negotiating commitments and developing the master test plan. The test manager, directly or by delegation, co-ordinates the project's master test planning process.



**GP 2.5 Train people**

*Train the people performing or supporting the test lifecycle and integration process as needed.*

**Elaboration**

Individuals involved in establishing the set of organizational test process assets are trained in developing and maintaining processes.

*Examples of topics include the following:*

- TMMi, CMMI and other (test) process reference models
- Planning, managing and monitoring processes
- Process modeling and definition
- Developing a tailorable standard process
- Developing work environment standards
- Ergonomics

Test management, and other individuals or groups involved, are trained in master test planning and the accompanying procedures and techniques.

*Examples of topics include the following:*

- Planning principles
- Test strategy
- Product and project risk assessment process and techniques
- Defining a test approach
- Test plan templates and standards
- Organizational structures
- Test estimation and test scheduling
- Supporting test planning tools

**GP 2.6 Manage configurations**

*Place designated work products of the test lifecycle and integration process under appropriate levels of configuration control.*

**Elaboration**

*Examples of work products placed under configuration management include the following:*

- Organization's set of standard test processes
- Description of integrated test lifecycle models
- Tailoring guidelines for the organization's set of standard test processes
- Organization's test process and product quality measurement data
- Work breakdown structure
- Test estimation data
- Product risk assessment data
- Master test plan review report
- Master test plan

**GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the test lifecycle and integration process as planned.*

**Elaboration**

*Examples of activities for stakeholder involvement include the following:*

- Reviewing the organization's set of standard test processes
- Reviewing the organization's integrated lifecycle models
- Resolving issues of tailoring guidelines
- Assessing the definitions of the common set of test process and product quality measurement data
- Reviewing the work environment standards



- Selecting the product and product components to be tested
- Identifying the risk level and risk types of the product and product components to be tested by being involved in a product risk assessment
- Providing input to test estimates
- Reviewing and resolving issues on test project risks
- Explicitly committing to test resources needed
- Reviewing and approval of master test plan

## GP 2.8 Monitor and control the process

*Monitor and Control the test lifecycle and integration process against the plan for performing the process and take appropriate actions.*

### Elaboration

Execution of the master test plan is typically monitored and controlled by means of the practices of the process area "Test Monitoring and Control".

*Examples of measures used in monitoring and control the test lifecycle and integration process include the following:*

- Percentage of projects using the test process elements of the organization's set of standard test processes
- Number of change requests for each test process element of the organization's set of standard test processes
- Number of staff member's compensation claims due to ergonomic problems
- Schedule for development of a test process or test process change
- Number of revisions to the master test plan
- Effort spent and actual lead-time compared to the effort planned and planned lead-time in the master test plan
- Number of test items' risk level changed per revision
- Cost, schedule and effort variance per plan revision

## GP 2.9 Objectively evaluate adherence

*Objectively evaluate adherence of the test lifecycle and integration process against its process description, standards, and procedures, and address non-compliances.*

### Elaboration

*Examples of review and/or audit evaluation adherence topics include the following:*

- Activities for establishing organizational test process assets
- Organization's set of standard test processes
- Description of test lifecycle models
- Tailoring guidelines for the organization's set of standard test processes
- Organization's test process data
- Compliance to the test strategy
- Compliance to standards (procedures and templates)
- Quality of the master test plan
- Defined test approach
- Product risk assessment process
- Test estimation process
- Activities for reviewing and making test commitments

## GP 2.10 Review status with higher level management

*Review the activities, status and results of the test lifecycle and integration process with higher level management and resolve issues.*

## GG 3 Institutionalize a Defined Process

### GP 3.1 Establish a defined process

*Establish and maintain a description of a defined test lifecycle and integration process*

### GP 3.2 Collect improvement information

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the test lifecycle and integration process to support the future use and improvement of the organization's processes and process assets.*

#### **Elaboration**

*Examples of measures include the following:*

- Submission of lessons learned to the organization's process asset library
- Submission of data to the organization's test process database
- Status of change requests submitted to modify the organization's standard process
- Percentage of master test plans established according to procedure and template
- Percentage of master test plans that have documented product risk assessment results and a test approach
- Percentage of master test plans formally reviewed and approved by management
- Master test planning effort
- Test estimation accuracy

## PA 3.4 Non-Functional Testing

### Purpose

The purpose of non-functional testing is to improve test process capability for non-functional testing during test planning, test design and execution by defining a test approach based on the identified non-functional product risks, establishing non-functional test specifications and executing a structured test execution process focused on non-functional testing.

### Introductory notes

Quality of products is all about satisfying stakeholders' needs. These needs have to be translated to well-described functional and non-functional requirements. Often the non-functional requirements are highly important for customer satisfaction. This process area addresses the development of a capability for non-functional testing. There is a set of principal non-functional attributes that are used to describe the quality of software products or systems. These quality attributes can be assessed using non-functional test techniques. Application of various test techniques will vary amongst others depending on the ability of the tester, the knowledge of the domain, and the attributes being addressed.

A test approach needs to be defined based on the outcome of a non-functional product risk assessment. Depending on the level and type of non-functional risks, it is decided which requirements of the product will be tested, to what degree and how. The non-functional product risks and test approach are defined in close co-operation between test specialists and the stakeholders. Testers should not take these decisions themselves.

Non-functional test techniques are applied, possibly supported by tools. Test techniques are used to derive and select non-functional test conditions and subsequently test cases from non-functional requirements and design specifications. The test cases are subsequently translated into manual test procedures and/or automated test scripts. Specific test data required to be able to run the non-functional test is created. During the test execution stage, the non-functional tests will be executed, incidents are found and incident reports are written.

### Scope

The process area Non-Functional Testing involves performing a non-functional product risk assessment and defining a test approach based on the non-functional risks identified. It also addresses the test preparation phase to derive and select non-functional test conditions and test cases, the creation of specific test data and the execution of the non-functional tests. Test environment practices, which are often critical for non-functional testing, are not addressed within this process area. They are addressed as part of the TMMi level 2 process area "Test Environment".

### Specific Goal and Practice Summary

- SG1 Perform a non-functional product risk assessment
  - SP 1.1 Identify non-functional product risks
  - SP 1.2 Analyze non-functional product risks
- SG2 Establish a non-functional test approach
  - SP 2.1 Identify features to be tested
  - SP 2.2 Define the non-functional test approach
  - SP 2.3 Define non-functional exit criteria
- SG3 Perform non-functional test analysis and design
  - SP 3.1 Identify and prioritize non-functional test conditions
  - SP 3.2 Identify and prioritize non-functional test cases
  - SP 3.3 Identify necessary specific test data
  - SP 3.4 Maintain horizontal traceability with non-functional requirements
- SG4 Perform non-functional test implementation
  - SP 4.1 Develop and prioritize non-functional test procedures
  - SP 4.2 Create specific test data
- SG5 Perform non-functional test execution
  - SP 5.1 Execute non-functional test cases
  - SP 5.2 Report non-functional test incidents
  - SP 5.3 Write test log

## Specific Practices by Goals

### SG 1 Perform a Non-Functional Product Risk Assessment

*A product risk assessment is performed to identify the critical areas for non-functional testing.*

#### SP 1.1 Identify non-functional product risks

*Non-functional product risks are identified and documented.*

##### **Typical work products**

1. Identified non-functional product risks

##### **Sub-practices**

1. Identify and select stakeholders that need to contribute to the risk assessment
2. Identify non-functional product risks based on defined non-functional product risk categories using input from stakeholders and requirements documents

*Examples of product risk identification techniques include the following:*

- Risk workshops
- Brainstorming
- Expert interviews
- Checklists
- Lessons learned

3. Document the context-of-use and potential effects of the non-functional risk
4. Identify the relevant stakeholders associated for each non-functional risk

Note that in practice the identification of non-functional products risk may be combined with SP 1.2 “Identify products risks” of the “Test Planning” process area and/or with the SP 3.1 “Perform a product risk assessment” of the “Test Lifecycle and Integration process area when establishing a master test plan.

#### SP 1.2 Analyze non-functional product risks

*Non-functional product risks are evaluated, categorized and prioritized using predefined categories and parameters.*

##### **Typical work products**

1. Non-functional product risk list, with a category and priority assigned to each risk

##### **Sub-practices**

1. Analyze the identified non-functional products risks using the predefined parameters, e.g. likelihood and impact
2. Categorize and group non-functional product risks according to the defined risk categories

*Examples of non-functional risks categories include the following [ISO 9126]:*

- Usability
- Reliability
- Efficiency
- Maintainability
- Portability

3. Prioritize the non-functional product risks for mitigation
4. Establish traceability between non-functional products risks and requirements to ensure that the source of product risks is documented
5. Generate a non-functional requirements / product risks traceability matrix
6. Review and obtain agreement with stakeholders on the completeness, category and priority level of the non-functional product risks
7. Revise the non-functional product risks as appropriate

*Examples of when product risks may need to be revised include the following:*

- New or changing non-functional requirements
- Change of the software development approach
- Lessons learned on quality issues in the project

Note that product risk categories and parameters as defined in the “Test Planning” process area (SP 1.1 “Define product risk categories and parameters”) are largely re-used and potentially also enhanced within this and the next specific practice.

## **SG 2** Establish a Non-Functional Test Approach

*A test approach for non-functional testing, based on identified non-functional product risks, is established and agreed upon.*

### **SP 2.1** Identify non-functional features to be tested

*The non-functional features to be tested, and not to be tested, are identified based on the product risks.*

#### **Typical work products**

1. List of non-functional features to be tested and not to be tested

#### **Sub-practices**

1. Breakdown the prioritized non-functional product risks into non-functional features to be tested and not to be tested
2. Document the risk level and source documentation (test basis) for each identified feature to be tested

### **SP 2.2** Define the non-functional test approach

*The test approach is defined to mitigate the identified and prioritized non-functional product risks.*

#### **Typical work products**

1. Non-functional test approach (documented in a test plan)

The approach should be described in sufficient detail to support the identification of major test tasks and estimation of the time required to do each one.

#### **Sub-practices**

1. Select the non-functional test techniques to be used

*Examples of non-functional test techniques to be selected include the following:*

- Heuristic evaluation, survey and questionnaires for usability
- Operational profiles for reliability
- Load, stress and volume testing for efficiency
- Checklists

Note that also often also black box, white box techniques and experienced-based techniques such as exploratory testing can be selected to test specific non-functional quality attributes.

2. Define the approach to reviewing test work products
3. Define the approach for non-functional re-testing
4. Define the approach for non-functional regression testing
5. Define the supporting test tools to be used
6. Identify significant constraints regarding the non-functional test approach, such as test resource availability, test environment features and deadlines
7. Align the non-functional test approach with the defined organization-wide or program-wide test strategy
8. Identify any non-compliances to the test strategy and its rationale
9. Review the non-functional test approach with stakeholders
10. Revise the non-functional test approach as appropriate

*Examples of when the non-functional test approach may need to be revised include the following:*

- New or changed priority level of non-functional product risks

- Lessons learned on applying the non-functional test approach in the project

### SP 2.3 Define non-functional exit criteria

*The exit criteria for non-functional testing are defined to plan when to stop testing.*

#### Typical work products

1. Non-functional exit criteria

#### Sub-practices

1. Define a set of non-functional exit criteria related to the product quality attributes to be tested

*Examples of non-functional exit criteria related to product quality attributes include the following:*

- For reliability: Mean Time Between Failures (MTTF), Mean Time to Repair (MTTR)
- For usability: user satisfaction, average time to perform functions
- For efficiency: mean response time, memory utilization
- For maintainability: average effort to change, availability of documentation

2. Review the non-functional exit criteria with stakeholders

Note that exit criteria of a test level should be aligned with the entry criteria of a subsequent test level.

Note that entry, suspension and resumption criteria are not explicitly defined within this process area by means of specific practices. In practices the defined entry, suspension and resumption criteria as part of the process area 'Test Planning' are largely also applicable to non-functional testing. E.g. entry criteria such as the availability of a test environment, a successful intake test and the availability of a test release note are applicable to all types of testing, both functional and non-functional.

### SG 3 Perform Non-Functional Test Analysis and Design

*During test analysis and design the test approach for non-functional testing is translated into tangible test conditions and test cases.*

#### SP 3.1 Identify and prioritize non-functional test conditions.

*Test conditions are identified and prioritized, based on an analysis of non-functional test features as specified in the test basis*

#### Typical work products

1. Test basis issue log
2. Non-functional test conditions
3. Non-functional test design specification

#### Sub-practices

1. Study and analyze the test basis (such as non-functional requirements, architecture, design and interface specifications)
2. Discuss issues regarding the test basis with the document owner
3. Derive the test conditions from the test basis in line with the documented non-functional test approach
4. Prioritize the test conditions based on identified product risks
5. Document the test conditions in a test design specification, based on the test design specification standard

*Examples of elements of a test design specification include the following [after IEEE 829]:*

- Test design specification identifier
- Features (and/or items) to be tested
- Approach refinements
- Test conditions
- Pass/fail criteria

6. Review the test design specifications with stakeholders
7. Revise the test design specifications and test conditions as appropriate, e.g. whenever the requirements change.

**SP 3.2 Identify and prioritize non-functional test cases**

*Non-functional test cases are identified and prioritized.*

**Typical work products**

1. Non-functional test cases
2. Non-functional test case specification

**Sub-practices**

1. Derive the test cases from the test conditions in line with the documented non-functional test approach.
2. Prioritize the test cases based on identified non-functional product risks
3. Document the non-functional test cases in a test case specification, based on the test case specification standard

*Examples of elements of a test case specification include the following [IEEE 829]:*

- Test case specification identifier
- Features (and/or items) to be tested
- Input specifications
- Output specifications
- Environmental needs
- Special procedural requirements
- Inter-case dependencies

4. Review the test case specifications with stakeholders
5. Revise the test case specifications as appropriate

**SP 3.3 Identify necessary specific test data**

*Specific test data necessary to support the non-functional test conditions and test cases is identified*

**Typical work products**

1. Test data specification

**Sub-practices**

1. Identify and specify the necessary specific test data required to implement and execute the non-functional test cases
2. Document the necessary specific test data, possibly as part of the test case specification

**SP 3.4 Maintain horizontal traceability with non-functional requirements**

*Maintain horizontal traceability from non-functional requirements to non-functional test conditions.*

**Typical work products**

1. Non-functional requirements / test conditions traceability matrix

**Sub-practices**

1. Maintain non-functional requirements traceability to ensure that the source of non-functional test conditions is documented
2. Generate a non-functional requirements / test conditions traceability matrix
3. Set up the traceability matrix such that monitoring of non-functional requirements coverage during test execution is facilitated

**SG 4 Perform Non-Functional Test Implementation**

*Non-functional test procedure are developed and prioritized, and specific test data required for non-functional testing is created.*

**SP 4.1 Develop and prioritize non-functional test procedures**

*Non-functional test procedures are developed and prioritized.*



**Typical work products**

1. Non-functional test procedure specification
2. Automated test script

**Sub-practices**

1. Develop non-functional test procedures by combining the non-functional test cases in a particular order and including any other information needed for test execution
2. Prioritize the non-functional test procedures based on identified product risks
3. Document the non-functional test procedures in a test procedure specification, based on the test procedure specification standard

*Examples of elements of a test procedure specification include the following [IEEE 829]:*

- Test procedure specification identifier
- Purpose
- Special requirements (execution preconditions)
- Procedure steps (test actions and checks)

4. Review the non-functional test procedure specifications with stakeholders
5. Revise the non-functional test procedure specifications as appropriate
6. Optionally, the non-functional test procedures can be automated and translated into an automated test scripts, e.g. for endurance testing or performance testing
7. Schedule the non-functional test procedures as part the overall test execution schedule

Refer to SP 2.4 “Develop test execution schedule” from the process area “Test Design and Execution” for scheduling the execution of test procedures and test scripts.

**SP 4.2 Create specific test data**

*Specific test data to support the non-functional testing as specified during the test analysis and design activity is created.*

**Typical work products**

1. Test data

**Sub-practices**

1. Create specific test data required to perform the non-functional tests as specified in the test procedures
2. Archive the set of specific test data to allow a restore of the initial situation in the future

Refer to SP 3.2 “Perform test data management” from the process area “Test Environment” for managing the created test data.

**SG 5 Perform Non-Functional Test Execution**

*Non-functional tests are executed in line with previously specified test procedures. Incidents are reported and test logs are written.*

**SP 5.1 Execute non-functional test cases**

*The non-functional test cases are executed manually using documented test procedures and/or or automated using test scripts.*

**Typical work products**

1. Test results

**Sub-practices**

1. Execute the non-functional test cases using document test procedures and/or test scripts
2. Record actual results
3. Compare actual results with expected results
4. Repeat non-functional test activities as a result of an action for an incident found by performing re-testing (confirmation testing)
5. Perform non-functional regression testing as appropriate.

Note that some non-functional testing will be carried informally using no pre-defined detailed test procedures, e.g. a heuristic evaluation to test the usability.

Note that the non-functional test execution is normally preceded by the overall intake test. Refer to the practices SP 2.3 “Specify intake test procedure” and SP 3.1 “Perform intake test” from the process area Test Design and Execution for more details on the intake test on the test object and to the practice SP2.4 “Perform test environment intake test” from the process area “Test Environment” for more details on the intake test on the test environment.

## SP 5.2 Report test incidents

*Discrepancies are reported as non-functional test incidents when there are differences between actual and expected results.*

### Typical work products

1. Non-functional test incident reports

### Sub-practices

1. Log non-functional test incidents when a discrepancy is observed.
2. Analyze the non-functional test incident for further information on the problem
3. Establish the cause of the non-functional test incident, e.g. system under test, test documentation, test data or test execution mistake
4. Assign an initial priority and severity level to the non-functional test incident
5. Formally report the test incident using an incident classification scheme

*Examples of elements of a test incident report include the following [IEEE 829]:*

- Test incident report identifier
- Summary
- Incident description (input, expected results, actual results, anomalies, date and time, test procedure step, environment, attempts to repeat, testers, observers)
- Priority level
- Severity level

6. Review the non-functional test incident report with stakeholders
7. Non-functional test incidents are stored in a central repository

Refer to the goal SG4 “Manage test incidents to closure” from the process area “Test Design and Execution” for more details on how test incidents are processed and managed to closure.

## SP 5.3 Write test log

*Test logs are written to provide a chronological record of relevant details about the execution of the non-functional tests.*

### Typical work products

1. Test logs

### Sub-practices

1. Collect test execution data
2. Document the test execution data by means of a test log, based on the test log standard

*Examples of elements of a test log include the following [IEEE 829]:*

- Test log identifier
- Description (items being tested, environment in which the testing has been executed)
- Activity and event entries (execution description, test results, anomalous events, incident report identifiers)

3. Review the test log with stakeholders

## Generic Practices by Goals

### GG 2 Institutionalize a Managed Process

#### GP 2.1 Establish an organizational policy

*Establish and maintain an organizational policy for planning and performing the non-functional testing process.*

##### **Elaboration**

The policy for non-functional testing typically specifies:

- Typical quality attributes that are important to the business and products
- A set of important and relevant quality attributes per test level
- The level of test automation and type of tools required
- Non-functional test incidents are documented and reported using an incident classification scheme
- Reported non-functional test incidents are evaluated, classified and processed according to a documented procedure

#### GP 2.2 Plan the process

*Establish and maintain the plan for performing the non-functional testing process.*

##### **Elaboration**

Typically, the plan for performing the non-functional testing process is included in the test plan, which is described in the TMMi “Test Planning” process area. The activities for non-functional testing are explicitly scheduled as part of the test plan.

#### GP 2.3 Provide resources

*Provide adequate resources for performing the non-functional testing process, developing the test work products, and providing the services of the process.*

##### **Elaboration**

- Adequate time is provided to perform the non-functional test planning, design and execution activities
- Experienced individuals, who have expertise in non-functional testing activities and tools are available
- Experienced individuals, who have expertise in the application domain of the test object and those who have expertise on the development process are available to support the development of the non-functional test approach, e.g. participating in product risk-analysis sessions, and the non-functional test designs, e.g. participating at reviews
- Tools to support the non-functional test design and execution process are available

*Examples of tools include the following:*

- Monitoring tool
- Performance tool
- Static analysis tool
- Dynamic analysis tool

#### GP 2.4 Assign responsibilities

*Assign responsibility and authority for performing the non-functional testing process, developing the work products, and providing the services of non-functional testing process.*

#### GP 2.5 Train people

*Train the people performing or supporting the non-functional testing process as needed.*

##### **Elaboration**

Test specialists, and other individuals or groups, involved in non-functional testing, are trained in non-functional testing and the accompanying procedures, techniques and tools.

*Examples training topics include the following:*

- The importance of non-functional testing
- Quality attributes (e.g. ISO 9126)
- Product risk-analysis for non-functional testing
- Defining a test approach for non-functional testing
- Formal and informal test techniques for non-functional testing
- Exit criteria for non-functional attributes
- Supporting tools

## **GP 2.6** Manage configurations

*Place designated work products of the non-functional testing process under appropriate levels of configuration control.*

### **Elaboration**

*Examples of work products placed under configuration management include the following:*

- Non-functional product risk assessment data
- Non-functional test design specifications
- Non-functional test case specifications
- Non-functional test procedure specifications (and/or test scripts)
- Test logs

## **GP 2.7** Identify and involve relevant stakeholders

*Identify and involve relevant stakeholders of the non functional testing process as planned.*

### **Elaboration**

*Examples of activities for stakeholder involvement include:*

- Identifying the non-functional risks of the product and product components to be tested by being involved in a product risk assessment
- Reviewing and approval of non-functional test designs and test cases
- Executing tests, e.g. usability testing by end-users

## **GP 2.8** Monitor and control the process

*Monitor and control the non-functional testing process against the plan for performing the process and take appropriate actions.*

### **Elaboration**

*Examples of measures used to monitor and control the non-functional testing process include the following:*

- Number of non-functional test specifications completed
- Number of non-functional tests executed
- Number of non-functional risks mitigated
- Number of outstanding non-functional incidents (per priority level)

## **GP 2.9** Objectively evaluate adherence

*Objective evaluate adherence of the non-functional testing process against its process description, standards, and procedures, and address non-compliances.*

### **Elaboration**

*Examples of review and/or audit evaluation adherence topics include the following:*

- Compliance to the non-functional aspects of the test strategy
- The defined test approach for non-functional testing
- The non-functional product risk assessment process
- The effectiveness and efficiency of non-functional test design techniques

- The quality of the non-functional test cases

#### **GP 2.10 Review status with higher level management**

*Review the activities, status and results of non-functional testing process with higher level management and resolve issues.*

### **GG 3 Institutionalize a Defined Process**

#### **GP 3.1 Establish a defined process**

*Establish and maintain a description of a defined non-functional testing process.*

#### **GP 3.2 Collect improvement information**

*Collect work products, measures, measurement results, and improvement information derived from planning and performing the non-functional testing process to support the future use and improvement of the organization's processes and process assets.*

*Examples of measures include the following:*

- Effort ratio of non-functional testing vs. functional testing
- Test effort spent per non-functional attribute
- Number of non-functional attributes tested per project
- Non-functional incident reports by priority and severity
- Non-functional requirements coverage achieved

## PA 3.5 Peer Reviews

### Purpose

The purpose of Peer Reviews is to verify that work products meet their specified requirements and to remove defects from selected work products early and efficiently. An important corollary effect is to develop a better understanding of the work products and of defect that might be prevented.

### Introductory notes

Reviews involve a methodical examination of work products by peers to identify defects and areas where changes are needed. Reviews are carried out through a small group of engineers, generally between 2-7 persons. The work product to be reviewed could be a requirements specification, design, source code, test design, a user manual, or other documents. In practice, there are many ways by which the group of reviewers actually gets chosen. Reviewers may be:

- Specialists in reviewing (quality assurance or audit)
- People from the same project
- People invited by the author because of their specific knowledge
- People, e.g. business representatives, because of a significant interest in the product.

Several types of reviews exist, each with its own purpose and objective. In addition to informal reviews, more formal review types are walkthroughs, technical reviews and inspections [IEEE 1028]. In a walkthrough, the author guides a group of people through a document and his thought process, so everybody understands the document in the same way and they reach consensus on the content or changes to be made. In a technical review the group discusses, after an individual preparation, the content and the (technical) approach to be used. An inspection, the most formal review type, is a technique where a document is checked for defects individually and in a group, using sources and standards, according to rules.

### Scope

The Peer Review process area covers the practices for performing peer reviews on work product, e.g. by testers reviewing a requirements specification for testability. It also includes the practices for establishing the peer review approach within a project. Project reviews (management reviews) are outside the scope of this process areas. At TMMi maturity level 3 peer reviews are not yet fully integrated with the dynamic testing process, e.g. part of the test strategy, test plan and test approach.

### Specific Goal and Practice Summary

- SG 1 Establish a peer review approach
  - SP 1.1 Identify work products to be reviewed
  - SP 1.2 Define peer review criteria
- SG2 Perform peer reviews
  - SP 2.1 Conduct peer reviews
  - SP 2.2 Testers review the test basis documents
  - SP 2.3 Analyze peer review data

### Specific Practices by Goals

#### SG 1 Establish a Peer Review Approach

*A review approach is established and agreed upon.*

#### SP 1.1 Identify work product to be reviewed

*The work products to be reviewed are identified, including the type of review and critical participants (stakeholders) to involve.*

##### **Typical work products**

1. List of work products to be reviewed
2. Review approach review log
3. Peer review schedule

### **Sub-practices**

1. Select work products that will undergo a peer review based on the peer review policy and the identified product risks
2. Determine what types of peer review will be conducted for the selected work products

*Examples of types of peer reviews include the following (IEEE 1028):*

- Inspections
- Walkthrough
- Technical Review
- Informal review

Note, it is possible that multiple types of reviews are selected for the same work product, e.g. for work products related to critical product risks.

3. Identify key participants that shall be involved in a peer review
4. Review the defined review approach with stakeholders
5. Develop a detailed peer review schedule, including the dates for peer review training and when material for peer reviews will be available
6. Obtain commitments to undertake the review approach and schedule from key stakeholders

## **SP 1.2 Define peer review criteria**

*Prepare for peer reviews on selected work products.*

### **Typical work products**

1. Peer review entry and exit criteria
2. Criteria for requiring another peer review

### **Sub-practices**

1. Establish and maintain entry criteria for peer reviews

*Examples of peer reviews entry criteria include the following:*

- A short cursory check of a product sample by the review leader (or an expert) does not reveal a large number of major defects
- The document has been cleaned up by a spelling-checker or other computer "diagnosis"
- References needed for inspection are up to date and available
- All source (i.e. higher level) documents shall have exited their own review
- The document author is prepared to join the inspection and feels confident regarding the quality of the document

Note that entry criteria will differ depending on the type of review that will be performed.

2. Establish and maintain exit criteria for peer reviews

*Examples of peer reviews exit criteria include the following:*

- Number of major defects found per page
- Preparation time spent as agreed upon
- All pages checked according to plan
- All issues and action item addressed
- Estimated residual defect density

3. Establish and maintain criteria for requiring another peer review
4. Review the defined criteria with stakeholders

## **SG 2 Perform Peer Reviews**

*Peer reviews are performed on selected work products and peer review data is analyzed.*

### **SP 2.1 Conduct peer reviews**

*Conduct peer reviews on selected work products and identify issues resulting from the peer reviews.*

#### **Typical work products**

1. Peer review logging forms (defects found)



2. Peer review action item
3. Peer review data (e.g. documented on process forms)
4. Peer review report (e.g. documented on process form)

#### **Sub-practices**

1. Ensure that the work product satisfies the peer review entry criteria prior to distribution
2. Distribute the work product to be reviewed and its related information to participants early enough to enable participants to adequately prepare for peer review
3. Assign individuals to roles for the peer review as appropriate

*Examples of roles include the following:*

- Review Leader (Moderator)
- Checker (Reviewer)
- Scribe
- Author

4. Perform the assigned roles in the peer review
5. Identify and document defects and other issues in the work product
6. Record the results of the peer review, e.g. on logging forms
7. Identify action items and communicate the issues to relevant stakeholders
8. Conduct an additional peer review if the defined criteria indicate the need
9. Ensure that the exit criteria for the peer review are satisfied
10. Record peer review data related to the preparation, conduct, and results of the peer reviews

*Typical data are product type, product size, type of peer review, number of reviewers, preparation time per reviewer, length of the review meeting, number of (major) defects found etc.*

### **SP 2.2 Testers review the test basis documents**

*The documents that are used as a basis for testing are reviewed by the testers.*

#### **Typical work products**

1. Testability defects
2. Testability review report

#### **Sub-practices**

1. Testers review the test basis documents for testability, e.g. whether the chosen test design techniques can be applied on the test basis
2. Defects found during the review of the test basis documents are logged and reported
3. Test basis documents are improved based on the defects reported by testing

### **SP 2.3 Analyze peer review data**

*Analyze data about preparation, conduct, and results of the peer reviews.*

#### **Typical work products**

1. Peer review database
2. Peer review analysis communication report

#### **Sub-practices**

1. Store the peer review data for future reference and analysis
2. Protect the review data to ensure that it is not used inappropriately

*Examples of inappropriate use of peer reviews include using data to evaluate the performance of people and using data for attribution.*

3. Analyze the peer review data

*Examples of peer review data that can be analyzed include the following:*

- Phase defect was injected
- Preparation effort or rate versus expected effort or rate
- Actual review effort versus planned review effort
- Number of defects versus number expected

- Types and severity level of defects detected
- Causes of defects
- Defect resolution impact

4. Communicate peer review analysis results to stakeholders

## Generic Practices by Goals

### GG 2 Institutionalize a Managed Process

#### GP 2.1 Establish an organizational policy

*Establish and maintain an organizational policy for planning and performing the peer review process.*

##### **Elaboration**

The peer review policy typically specifies:

- Reviews will be applied to identify defects early in the development lifecycle
- The organization identifies a standard set of work products that will undergo review, including test deliverables
- Each test project selects the work products that will undergo review and the associated review type(s)
- Peer review leaders and other participants will be trained for their role
- Testers shall participate in reviews on development documents addressing testability issues

#### GP 2.2 Plan the process

*Establish and maintain the plan for performing the peer review process.*

##### **Elaboration**

At TMMi level 3, peer reviews are most often not a full part of the testing process. Therefore typically, the plan for performing the peer review process is included in the project plan, which is described in the CMMI Project Planning process area. Resources, e.g. review leaders, are explicitly planned for to allow the performance of peer reviews.

When at higher TMMi levels peer reviews become an integral part of the testing process, the plan for performing the peer review process is included in the (master) test plan.

#### GP 2.3 Provide resources

*Provide adequate resources for performing the peer reviews process, developing the test work products, and providing the services of the process.*

##### **Elaboration**

- Meeting rooms are available for review meeting
- Trained peer review leaders are available
- Supporting artifacts such as defect logging forms and review process forms to support data collection and reporting are available
- Establish and maintain checklists, e.g. on testability, to ensure that the work products are reviewed consistently

*Examples of items addressed by the checklists include the following:*

- Compliance with standards
  - Design guidelines
  - Completeness
  - Correctness
  - Testability
  - Maintainability
  - Common defect types
- The checklists are modified as necessary to address the specific type of work product and peer review. The checklists are reviewed by peers and potential users of the checklists.

- Tools to support the peer review process are available, e.g. communication tools, data analysis tool and peer review process tool

#### **GP 2.4 Assign responsibilities**

*Assign responsibility and authority for performing the peer review process, developing the work products, and providing the services of the peer review process.*

##### **Elaboration**

Peer review leaders are designated to be responsible for co-coordinating the peer review process within projects.

#### **GP 2.5 Train people**

*Train the people performing or supporting the organizational test training process as needed.*

##### **Elaboration**

- Peer review leaders (moderators) receive training in how to lead peer reviews

*Examples of training topics for peer review leaders include the following:*

- Developing a peer review approach
- Type of reviews
- Peer review leader tasks and responsibility
- Leading and facilitating a meeting
- Achieving buy-in for reviews
- Peer review metrics

- Participants in peer reviews receive for their roles in the peer review process

*Examples of training topics for peer review participants include the following:*

- Objectives and benefits of peer reviews
- Types of reviews
- Peer review roles and responsibilities
- Peer review process overview
- Peer review preparation
- Document rules and checklists, e.g. regarding testability
- Peer review meetings

#### **GP 2.6 Manage configurations**

*Place designated work products of the peer review process under appropriate levels of configuration control.*

##### **Elaboration**

*Examples of work products placed under configuration management include the following:*

- Peer review approach
- Peer review logging and process forms
- Peer review data(base)
- Peer review training material

#### **GP 2.7 Identify and involve relevant stakeholders**

*Identify and involve relevant stakeholders of the peer review process as planned.*

##### **Elaboration**

Select relevant stakeholders from customers, end users, developers, testers, suppliers, marketers, maintenance, service, and other who may be affected by, or may affect, the (work) products to participate in peer reviews.

**GP 2.8 Monitor and control the process**

Monitor and Control the peer review against the plan for performing the process and take appropriate actions.

**Elaboration**

Examples of measures used in monitoring and control the peer review process include the following:

- Number of peer reviews planned and performed
- Number of work product reviewed compared to plan
- Number and type of defects found during peer reviews
- Schedule of peer review process activities (including training activities)
- Effort spent on peer reviews compared to plan

**GP 2.9 Objectively evaluate adherence**

Objective evaluate adherence of the peer review process against its process description, standards, and procedures, and address non-compliances.

**Elaboration**

Examples of review and/or audit evaluation adherence topics include the following:

- Verify whether peer reviews are performed
- Training on peer review leaders and other participants
- The process followed during peer reviews, including adherence to the defined criteria
- The actual performance on follow-up action items
- Peer review reporting regarding completeness on accuracy
- Peer review checklists used

**GP 2.10 Review status with higher level management**

Review the activities, status and results of the peer review process with higher level management and resolve issues.

**GG 3 Institutionalize a Defined Process****GP 3.1 Establish a defined process**

Establish and maintain a description of a defined peer review process

**GP 3.2 Collect improvement information**

Collect work products, measures, measurement results, and improvement information derived from planning and performing the peer review process to support the future use and improvement of the organization's processes and process assets.

**Elaboration**

Examples of measures include the following:

- Peer review data such as average preparation time, average total spent per peer review, average number of pages, etc.
- Number of defects found through peer reviews by development phase
- Return-On-Investment calculations

## 4 Comparing TPI and TMMi

In the testing industry there is, in addition to the TMMi, another leading model for test process improvement: the TPI<sup>5</sup> model. The TMMi and the Test Process Improvement (TPI) model developed by Sogeti share a common concern about the test process management. The two are driven by similar concerns and are intuitively correlated. The purpose of this chapter is to compare the TMMi (level 2) to the TPI model, identify their differences and similarities.

### 4.1 The TPI model

The TPI model [Koomen and Pol] gives practical guidelines for assessing the maturity level of testing in an organization and for step by step improvement of the process. The model consists of 20 key areas, each with different levels of maturity. The levels of all key areas are set out in a maturity matrix. Each level is described by several checkpoints. Improvement suggestions, which help to reach a desired level, are part of the model. The chapter includes a general description of the application of the model, which deals with how to implement and how to consolidate the improvements.

Based on the knowledge and experiences of a large number of professional testers the Test Process Improvement (TPI) model has been developed by Sogeti in The Netherlands. The TPI model supports the improvement of test processes. The model supports the determination of the “maturity” of the test processes within an organization. Based on this understanding the model helps to define gradual and controllable improvement steps. TPI is a more a continuous oriented model that is closely related to the structured test approach TMap [TMap].

The TPI model is visualized as follows:

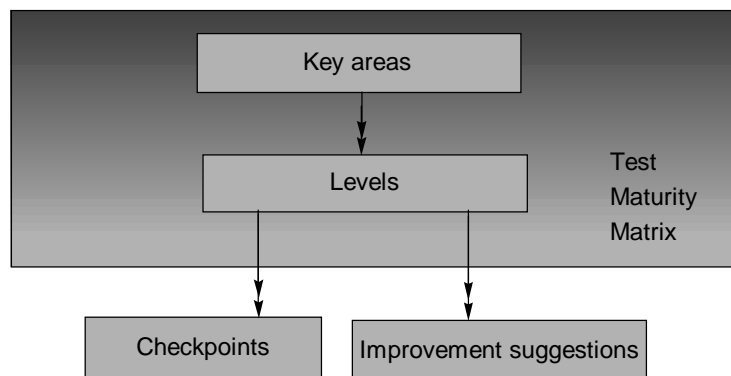


Figure 3: TPI model overview

#### 4.1.1 Key Areas

In each test process certain areas need specific attention in order to achieve a well-defined process. These key areas are therefore the basis for improving and structuring the test process. The key areas can be compared to the process areas of the TMMi. The TPI model has 20 key areas. Within the scope of TPI, test process improvement usually comprises mainly higher test levels like system and acceptance test. Most key areas are adjusted to this. However, to improve more “mature” test processes, attention must also be given to review activities and lower test levels like unit and integration tests. Separate key areas are included for these processes as well. The key areas are organized by means of the four cornerstones of structured testing as defined by TMap: lifecycle, organization, infrastructure and techniques.

The TPI key areas are:

##### Life Cycle

- Test Strategy
- Life-cycle Model
- Moment of Involvement

##### Techniques

- Estimating and Planning
- Test Specification Techniques
- Static Test Techniques
- Metrics

<sup>5</sup>TPI is a registered trademark of Sogeti, The Netherlands

**Infrastructure**

- Test Automation
- Test Environment
- Office Environment

**Organization**

- Commitment and Motivation
- Test Functions and Training
- Scope of Methodology
- Communication
- Reporting
- Defect Management
- Test ware Management
- Test Process Management

**All Cornerstones**

- Evaluation
- Low-level Testing

**4.1.2 Levels**

The way key areas are organized within the test process determines the “maturity”. It is obvious that not each key area will be addressed equally thoroughly; each test process has its strengths and weaknesses. In order to enable insight in the state of the key areas, the model supplies them with levels (from A to B to C to D). On the average, there are three levels for each key area. Each higher level (C being higher than B, B being higher than A) indicates a higher level of maturity. By using levels one can assess the current situation of the test process. It also increases the ability to define targets for stepwise improvement.

Each level consists of a number of requirements for the key area. The requirements (checkpoints) of a certain level also comprise the requirements of a lower level; a key area at level B fulfils the requirements of both level A and B. If a key area does not satisfy the requirements for level A, it is considered to be at the lowest and, consequently, undefined level for that particular key area. In the table hereafter an overview is provided of the different levels of the key areas.

<b>Levels Key areas</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
1. Test strategy	Strategy for single high-level test	Combined strategy for high-level tests	Combined strategy for high-level tests plus low-level tests or evaluation	Combined strategy for all test and evaluation levels
2. Life-cycle model	Planning, Specification, Execution	Planning, Preparation, Specification, Execution, Completion		
3. Moment of involvement	Completion of test basis	Start of test basis	Start of requirements definition	Project initiation
4. Estimating and planning	Substantiated estimating and planning	Statistically substantiated estimating and planning		
5. Test specification techniques	Informal techniques	Formal techniques		
6. Static test techniques	Inspection of test basis	Checklists		
7. Metrics	Project metrics (product)	Project metrics (process)	System metrics	Organization metrics (>1 system)
8. Test automation	Use of tools	Managed test automation	Optimal test automation	

Levels Key areas	A	B	C	D
9. Test environment	Managed and controlled environment	Testing in most suitable environment	Environment on demand	
10. Office environment	Adequate and timely office environment			
11. Commitment and motivation	Assignment of budget and time	Testing integrated in project organization	Test-engineering	
12. Test functions and training	Test manager and testers	(Formal) Methodical, technical and functional support, management	Formal internal Quality Assurance	
13. Scope of methodology	Project specific	Organization generic	Organization optimizing (R&D)	
14. Communication	Internal communication	Project communication (defects, change control)	Communication within the organization about the quality of the test processes	
15. Reporting	Defects	Progress (status of tests and products), activities (costs and time, milestones), defects with priorities	Risks and recommendations, substantiated with metrics	Recommendations have a Software Process Improvement character
16. Defect management	Internal defect management	Extended defect management with flexible reporting facilities	Project defect management	
17. Test ware management	Internal test ware management	External management of test basis and test object	Reusable test ware	Traceability system requirements to test cases
18. Test process management	Planning and execution	Planning, execution, monitoring, and adjusting	Monitoring and adjusting within organization	
19. Evaluation	Evaluation techniques	Evaluation strategy		
20. Low-level testing	Low-level test life-cycle: planning, specification and execution	White-box techniques	Low-level test strategy	

Table 4: TPI Levels

### 4.1.3 Checkpoints

In order to determine levels, the TPI model is supported by a measurement instrument. The requirements for each level are defined in the form of checkpoints; questions that need to be answered positively in order to classify for that level. Based on the checkpoints the test process can be assessed, and for each key area the maturity level can be established. As each next level of a key area is considered an improvement, this means that the checkpoints are cumulative. In order to classify for level B, one needs to answer positively to all checkpoints of both level B and level A.

### 4.1.4 Test Maturity Matrix

After determining the levels for each key area, attention should be directed as to which improvement steps to take. This is because not all key areas and levels are equally important. In addition to these priorities there are dependencies between the levels of different key areas. Before statistics can be gathered for defects found (level A of key area Metrics), the test process has to classify for level B of key area Defect management. Such dependencies can be found between many levels and key areas.



All levels and key areas are related to each other in the so-called Test Maturity Matrix. This has been done as a way to express the internal priorities and dependencies between levels and key areas. The vertical axis of the matrix indicates key areas; the horizontal axis shows scales of overall test process maturity. In the matrix each level is related to a certain scale of test maturity. This results in 13 scales of test maturity. The open cells between different levels have no meaning in themselves, but indicate that achieving a higher maturity for a key area is related to the maturity of other key areas. There is no gradation in between levels: as long as a test process is not entirely classified at level B, it remains at level A.

Scale Key area	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Test strategy		A					B				C		D	
2. Life-cycle model		A			B									
3. Moment of involvement			A				B				C		D	
4. Estimating and planning				A							B			
5. Test specification techniques		A		B										
6. Static test techniques					A		B							
7. Metrics						A			B			C		D
8. Test automation				A				B			C			
9. Test environment				A				B						C
10. Office environment				A										
11. Commitment and motivation		A				B						C		
12. Test functions and training				A			B			C				
13. Scope of methodology					A						B			C
14. Communication			A		B							C		
15. Reporting		A			B		C					D		
16. Defect management		A				B		C						
17. Test ware management			A			B				C				D
18. Test process management		A		B								C		
19. Evaluation							A			B				
20. Low-level testing					A		B		C					

Table 5: Test Maturity Matrix

The main purpose of the matrix is to show the strong and weak elements of the current test process and to support prioritization for improvement actions. The matrix works from left to right, so low mature key areas are improved first. As a consequence of the dependencies between levels and key areas, practice has shown that “outliners” (i.e., key areas with high scales of maturity, whereas surrounding key areas have medium or low scales) usually provide little return on investment.

### 4.1.5 Improvement Suggestions

Improvement actions can be defined in terms of desired higher levels. For reaching a higher level the checkpoints render much assistance. Beside these, the TPI model provides specific improvement Suggestions. Unlike the use of checkpoints, the use of improvement suggestions is not obligatory. Each level is supplied with several improvement suggestions.

## 4.2 Comparing TMMi to TPI

This section aims at briefly comparing the two test process improvement models. They are to different and therefore depending on the specific situation, one of the two models could be preferred.

Some differences between TMMi and TPI:

- Within TMMi all test levels are within scope. TPI focuses mainly on higher test levels.
- TMMi is strongly linked to CMMI. TPI is strongly linked to the TMap test methodology.
- TMMi strongly addresses management commitment very strongly and is therefore supporting a top-down improvement process. TPI is much more a bottom-up model that may be suitable to address test improvement for a specific (test) project or program. Some experts have stated that TMMi is a managerial model whereas TPI is an engineering model.
- TMMi addresses higher maturity testing issues such testability reviews, quality control, defect prevention and test measurement program. These issues are not or only partly addressed within TPI.

	<b>TPI</b>	<b>TMMi</b>
Type	Continuous model – Bottom-up oriented	Staged model – Top-down oriented
Test methods	Strongly related to TMap	Test method independent
SPI	No formal relationship to a SPI model	Highly correlated to CMMI
Test levels	Supports especially higher test levels	All test levels (including static testing)
Focus	20 Key areas	Detailed focus on limited number of process areas per maturity level
Approach	Test engineering	Also strong focus on management commitment

Table 6: TPI versus TMMi comparison

### 4.3 Mapping TMMi Level 2 to TPI

This section compares the 20 TPI key areas and its levels to the TMMi level 2 process areas with its specific and generic goals. The comparison, based on an analysis of the checkpoints (from TPI) and the practices and Sub-practices (from the TMMi), involves judgment, and so there are differences of interpretation for both TPI and the TMMi. A common challenge for assessments is reliability and consistency, which is partially addressed by strict prerequisites for TMMi assessment requirements and assessors. TPI does not provide such a scheme to address this issue. Note that also the terminology used by the model is often also different. TPI is based on the TMap terminology, whereas TMMi is based on the international testing terminology as defined by the ISTQB certification scheme [ISTQB].

	3.1 Test Policy and Strategy	3.2 Test Planning	2.3 Test Monitoring and Control	2.4 Test design and Execution	2.5 Test Environment
1 Test strategy	Levels B and C (combined strategy for multiple test levels) partly covered by SG2.	Level A (strategy for single test level) fully covered by SG1.			
2 Life-cycle model		Level A (planning, specification, execution) fully covered by SG3.			
3 Moment of involvement		Level A (completion of test basis) fully covered by SG2.			
4 Estimation & planning		Level A (substantiated estimation and planning) covered by SG1 and SG3.	Level A (substantiated estimation and planning) partly covered by SG3.		
5 Test specification techniques				Level A (informal techniques) and level B (formal techniques) fully covered by SG1.	

	3.1 Test Policy and Strategy	3.2 Test Planning	2.3 Test Monitoring and Control	2.4 Test design and Execution	2.5 Test Environment
6 Static test techniques				Level A (inspection of the test basis) partly covered by SG1.	
7 Metrics	Level B (project metrics [process]) covered by SG3.		Level A (project metrics [product]) fully covered by SG1 and SG2.	Level B (project metrics [process]) partly covered by SG2.	
8 Test automation		Level A (planning and control tools) fully covered by GG2 (GP2.3).	Level A (planning and control tools) fully covered by GG2 (GP2.3).	Level B (execution and analysis tool) fully covered by GG2 (GP2.3).	
9 Test environment					Level A (managed and controlled test environment) fully covered by various SG's. Level B (testing in the most suitable environment) partly covered.
10 Office environment		Level A (adequate and timely office environment) partly covered by SG3.	Level A (adequate and timely office environment) partly covered by SG1.		
11 Commitment & motivation	Level A (assignment of budget and time) partly covered by SG1.		Level A (assignment of budget and time) covered by GG2 (GP2.2 and GP 2.5)		
12 Test functions & training		Level A (test manager and testers) is partly covered by GG2 (GP2.3, GP2.4 and GP 2.5).	Level A (test manager and testers) is partly covered by GG2 (GP2.3, GP2.4 and GP 2.5).	Level A (test manager and testers) is partly covered by GG2 (GP2.3, GP2.4 and GP 2.5).	
13 Scope of methodology	At TMMi level 2, the test process is a managed process meaning that the process is at least instantiated at project level.				

	3.1 Test Policy and Strategy	3.2 Test Planning	2.3 Test Monitoring and Control	2.4 Test design and Execution	2.5 Test Environment
14 Communication			Level A (internal communication) and level B (project communication) partly covered by SG1 and SG2.	Level B (project communication) partly covered by SG4.	
15 Reporting			Level A (defects), level B (progress, activities, defects with priorities) and level C (risks and recommendations, substantiated with metric) are all fully covered by SG1 and SG2.		
16 Defect management			Level A (internal defect management), level B (extensive defect management) and level C (project defect management) are all fully covered by SG3, SG4 and GG2 (GP2.3 and GP2.4).		
17 Test ware management	Level A (internal test ware management) is fully covered by GG2 (GP 2.6 manage configurations). Level B (external management of test basis and test object) is not within the scope of the TMMi, but fully covered by the CMMI level 2 process areas Configuration Management. Level D (traceability system requirements to test cases) is fully covered by "Test Design and Execution" SG1 (SP1.4)				
18 Test Process management		Level A (planning and execution) is fully covered by SG3.	Level B (planning execution, monitoring and adjusting) is fully covered by SG1.		

	3.1 Test Policy and Strategy	3.2 Test Planning	2.3 Test Monitoring and Control	2.4 Test design and Execution	2.5 Test Environment
19 Evaluation	Although informal reviews are already applied at TMMi level 2, formal reviews are not part of TMMi level 2. They are part of TMMi level 3 (process area Peer Reviews).				
20 Low-level techniques		Level A (lifecycle model) fully covered by SG2; level C (low level test strategy) fully covered by SG1.		Level B (white-box techniques) fully covered by SG1.	

Table 7: Mapping TMMi level 2 to TPI

As a result of the mapping documented in the previous table, it is possible to establish a typical TPI test maturity matrix for a TMMi level 2 organizations. The typical test maturity matrix is documented hereafter. A TMMi level 2 organizations is at a minimum at TPI scale 3. However, for many key areas the TMMi level 2 organizations are already at scale 4, 5 or even beyond.

Scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Key area														
1. Test strategy		A					B				C		D	
2. Life-cycle model		A			B									
3. Moment of involvement			A				B				C		D	
4. Estimating and planning				A							B			
5. Test specification techniques		A		B										
6. Static test techniques					A		B							
7. Metrics						A		B	B			C		D
8. Test automation				A				B			C			
9. Test environment				A				B						C
10. Office environment				A										
11. Commitment and motivation		A				B						C		
12. Test functions and training				A			B			C				
13. Scope of methodology					A						B			C
14. Communication			A		B							C		
15. Reporting		A			B		C					D		
16. Defect management		A				B		C						
17. Test ware management			A			B				C				D
18. Test process management		A		B								C		
19. Evaluation							A			B				
20. Low-level testing					A		B		C					

Table 8: Typical TPI Test Maturity Matrix for a TMMi level 2 organization

### 4.4 Mapping TMMi Level 3 to TPI

This section compares the 20 TPI key areas and its levels to the TMMi level 3 process areas with its specific and generic goals. The comparison, based on an analysis of the checkpoints (from TPI) and the practices and Sub-practices (from the TMMi), involves judgment, and so there are differences of interpretation for both TPI and the TMMi.

3.1 Test Organization	3.2 Test Training Program	3.3 Test Lifecycle and Integration	3.4 Non-Functional Testing	3.5 Peer Reviews
-----------------------	---------------------------	------------------------------------	----------------------------	------------------

	3.1 Test Organization	3.2 Test Training Program	3.3 Test Lifecycle and Integration	3.4 Non-Functional Testing	3.5 Peer Reviews
1 Test strategy			Levels B and C (combined strategy for multiple test levels) fully covered by SG3.		
2 Life-cycle model	Level A (completion phase) covered by SG5.		Level B (preparation and completion phases) full covered by SG1.		Level B (preparation phase) covered by SG2.
3 Moment of involvement			Level B (completion of test basis) and Level C (start of requirements definition) fully covered by SG2		Level B (completion of test basis) covered by SG2.
4 Estimation & planning			Level B (statistically substantiated estimating and planning) covered by SG1 and SG3.		
5 Test specification techniques	Already fully covered at TMMi level 2				
6 Static test techniques				Level B (checklists) covered as being one of the techniques for non-functional testing.	Level A (inspection of the test basis) and B (checklist) covered by SG1.
7 Metrics			Level B (project metrics {process}) supported by SG1 (SP1.4).		
8 Test automation				Level B (execution and analysis tool) fully covered by GG2 (GP2.3).	

	3.1 Test Organization	3.2 Test Training Program	3.3 Test Lifecycle and Integration	3.4 Non-Functional Testing	3.5 Peer Reviews
9 Test environment			Level B (testing in the most suitable environment), the pre-condition of a coordinated test approach is now fully covered.		
10 Office environment			Level A (adequate and timely office environment) fully covered by SG1 (SP1.6)		
11 Commitment & motivation	Level B (test engineering) fully covered.		Level B (testing integrated in project organization) covered by SG2.		
12 Test functions & training	Level B (methodical, Technical, and Functional support) and Level C (formal Quality Assurance) fully covered.				
13 Scope of methodology	At TMMi level 3, the test process is a defined organization-wide standard process.				
14 Communication	Level C (communication within the organization about the quality of the test processes) fully covered by SG5.				
15 Reporting	Level D (recommendations have a software process improvement character) fully covered by SG5.				



	3.1 Test Organization	3.2 Test Training Program	3.3 Test Lifecycle and Integration	3.4 Non-Functional Testing	3.5 Peer Reviews
16 Defect management	Already fully covered at TMMi level 2				
17 Test ware management			Level C (re-usable test ware) is fully covered by SG1.	Level D (traceability system requirements to test cases) is fully covered by SG3(SP3.4)	
18 Test Process management	Level C (monitoring and adjustment within the organization) is fully covered by SG5.				
19 Evaluation				Level B (evaluation strategy) is fully covered by SG1.	
20 Low-level techniques	Already fully covered at TMMi level 2				

Table 9: Mapping TMMi level 3 to TPI

As a result of the mapping documented in the previous table, it is possible to establish a typical TPI test maturity matrix for a TMMi level 3 organizations. The typical test maturity matrix is documented hereafter. A TMMi level 3 organizations is usually at TPI scale 9 or 10.

Scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Key area</b>														
1. Test strategy		A					B				C		D	
2. Life-cycle model		A			B									
3. Moment of involvement			A				B				C		D	
4. Estimating and planning				A							B			
5. Test specification techniques		A		B										
6. Static test techniques					A		B							
7. Metrics						A			B			C		D
8. Test automation				A				B			C			
9. Test environment				A				B						C
10. Office environment				A										
11. Commitment and motivation		A				B						C		
12. Test functions and training				A			B			C				
13. Scope of methodology					A						B			C
14. Communication			A		B							C		
15. Reporting		A			B		C					D		
16. Defect management		A				B		C						
17. Test ware management			A			B				C				D
18. Test process management		A		B							C			

Scale Key area	0	1	2	3	4	5	6	7	8	9	10	11	12	13
19. Evaluation							A			B				
20. Low-level testing					A		B		C					

*Table 10: Typical TPI Test Maturity Matrix for a TMMi level 3 organization*

## 5 Glossary

<b>acceptance criteria</b>	The exit criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity. [IEEE 610]
<b>acceptance testing</b>	Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. [After IEEE 610]
<b>actual result</b>	The behavior produced/observed when a component or system is tested.
<b>alpha testing</b>	Simulated or actual operational testing by potential users/customers or an independent test team at the developers' site, but outside the development organization. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing.
<b>audit</b>	An independent evaluation of software products or processes to ascertain compliance to standards, guidelines, specifications, and/or procedures based on objective criteria, including documents that specify: (1) the form or content of the products to be produced (2) the process by which the products shall be produced (3) how compliance to standards or guidelines shall be measured. [IEEE 1028]
<b>best practice</b>	A superior method or innovative practice that contributes to the improved performance of an organization under given context, usually recognized as 'best' by other peer organizations.
<b>beta testing</b>	Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for off-the-shelf software in order to acquire feedback from the market.
<b>black-box testing</b>	Testing, either functional or non-functional, without reference to the internal structure of the component or system.
<b>black-box test design</b>	Technique/procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.
<b>boundary value analysis</b>	A black box test design technique in which test cases are designed based on boundary values.
<b>branch coverage</b>	The percentage of branches that have been exercised by a test suite. 100% branch coverage implies both 100% decision coverage and 100% statement coverage.
<b>branch testing</b>	A white box test design technique in which test cases are designed to execute branches.
<b>Capability Maturity Model (CMM)</b>	A five level staged framework that describes the key elements of an effective software process. The Capability Maturity Model covers best-practices for planning, engineering and managing software development and maintenance. [CMM]
<b>Capability Maturity Model Integration (CMMI)</b>	A framework that describes the key elements of an effective product development and maintenance process. The Capability Maturity Model Integration covers best-practices for planning, engineering and managing product development and maintenance. CMMI is the designated successor of the CMM. [CMMI]
<b>capture/playback tool</b>	A type of test execution tool where inputs are recorded during manual testing in order to generate automated test scripts that can be executed later (i.e. replayed). These tools are often used to support automated regression testing.
<b>cause-effect graphing</b>	A black box test design technique in which test cases are designed from cause-effect graphs. [BS 7925/2]

<b>classification tree method</b>	A black box test design technique in which test cases, described by means of a classification tree, are designed to execute combinations of representatives of input and/or output domains. [Grochtmann]
<b>code coverage</b>	An analysis method that determines which parts of the software have been executed (covered) by the test suite and which parts have not been executed, e.g. statement coverage, decision coverage or condition coverage.
<b>component</b>	A minimal software item that can be tested in isolation.
<b>component integration testing</b>	Testing performed to expose defects in the interfaces and interaction between integrated components.
<b>component testing</b>	The testing of individual software components. [After IEEE 610]
<b>condition coverage</b>	The percentage of condition outcomes that have been exercised by a test suite. 100% condition coverage requires each single condition in every decision statement to be tested as True and False.
<b>condition testing</b>	A white box test design technique in which test cases are designed to execute condition outcomes.
<b>configuration</b>	The composition of a component or system as defined by the number, nature, and interconnections of its constituent parts.
<b>configuration auditing</b>	The function to check on the contents of libraries of configuration items, e.g. for standards compliance. [IEEE 610]
<b>configuration control</b>	An element of configuration management, consisting of the evaluation, co-ordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. [IEEE 610]
<b>configuration control board (CCB)</b>	A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes. [IEEE 610]
<b>configuration identification</b>	An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. [IEEE 610]
<b>configuration item</b>	An aggregation of hardware, software or both, that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE 610]
<b>configuration management</b>	A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. [IEEE 610]
<b>configuration management tool</b>	A tool that provides support for the identification and control of configuration items, their status over changes and versions, and the release of baselines consisting of configuration items.
<b>continuous representation</b>	A capability maturity model structure wherein capability levels provide a recommended order for approaching process improvement within specified process areas. [CMMI]
<b>coverage tool</b>	A tool that provides objective measures of what structural elements, e.g. statements, branches have been exercised by a test suite.
<b>debugging tool</b>	A tool used by programmers to reproduce failures, investigate the state of programs and find the corresponding defect. Debuggers enable programmers to execute programs step by step, to halt a program at any program statement and to set and examine program variables.
<b>decision coverage</b>	The percentage of decision outcomes that have been exercised by a test suite. 100% decision coverage implies both 100% branch coverage and 100% statement coverage.

<b>decision table testing</b>	A black box test design technique in which test cases are designed to execute the combinations of inputs and/or stimuli (causes) shown in a decision table. [Veenendaal]
<b>decision testing</b>	A white box test design technique in which test cases are designed to execute decision outcomes.
<b>defect</b>	A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system.
<b>defect based test design technique</b>	A procedure to derive and/or select test cases targeted at one or more defect categories, with tests being developed from what is known about the specific defect category. See also defect taxonomy.
<b>defect density</b>	The number of defects identified in a component or system divided by the size of the component or system (expressed in standard measurement terms, e.g. lines-of-code, number of classes or function points).
<b>Defect Detection Percentage (DDP)</b>	The number of defects found by a test phase, divided by the number found by that test phase and any other means afterwards.
<b>defect management</b>	The process of recognizing, investigating, taking action and disposing of defects. It involves recording defects, classifying them and identifying the impact. [After IEEE 1044]
<b>defect management tool</b>	A tool that facilitates the recording and status tracking of defects and changes. They often have workflow-oriented facilities to track and control the allocation, correction and re-testing of defects and provide reporting facilities. See also incident management tool.
<b>defect masking</b>	An occurrence in which one defect prevents the detection of another. [After IEEE 610]
<b>defect report</b>	A document reporting on any flaw in a component or system that can cause the component or system to fail to perform its required function. [After IEEE 829]
<b>defect taxonomy</b>	A system of (hierarchical) categories designed to be a useful aid for reproducibly classifying defects.
<b>defined process</b>	A managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines; has maintained process description; and contributes work products, measures, and other process improvement information to the organizational process assets. [CMMI]
<b>deliverable</b>	Any (work) product that must be delivered to someone other than the (work) product's author.
<b>driver</b>	A software component or test tool that replaces a component that takes care of the control and/or the calling of a component or system. [After TMap]
<b>dynamic analysis tool</b>	A tool that provides run-time information on the state of the software code. These tools are most commonly used to identify unassigned pointers, check pointer arithmetic and to monitor the allocation, use and de-allocation of memory and to flag memory leaks.
<b>dynamic testing</b>	Testing that involves the execution of the software of a component or system.
<b>elementary comparison testing</b>	A black box test design technique in which test cases are designed to execute combinations of inputs using the concept of condition determination coverage. [TMap]
<b>emulator</b>	A device, computer program, or system that accepts the same inputs and produces the same outputs as a given system. [IEEE 610] See also simulator.
<b>entry criteria</b>	The set of generic and specific conditions for permitting a process to go forward with a defined task, e.g. test phase. The purpose of entry criteria is to prevent a task from starting which would entail more (wasted) effort compared to the effort needed to remove the failed entry criteria. [Gilb and Graham]

---

<b>equivalence partitioning</b>	A black box test design technique in which test cases are designed to execute representatives from equivalence partitions. In principle test cases are designed to cover each partition at least once.
<b>error</b>	A human action that produces an incorrect result. [After IEEE 610]
<b>error guessing</b>	A test design technique where the experience of the tester is used to anticipate what defects might be present in the component or system under test as a result of errors made, and to design tests specifically to expose them.
<b>exhaustive testing</b>	A test approach in which the test suite comprises all combinations of input values and preconditions.
<b>exit criteria</b>	The set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed. The purpose of exit criteria is to prevent a task from being considered completed when there are still outstanding parts of the task which have not been finished. Exit criteria are used to report against and to plan when to stop testing. [After Gilb and Graham]
<b>expected result</b>	The behavior predicted by the specification, or another source, of the component or system under specified conditions.
<b>experienced-based test design technique</b>	Procedure to derive and/or select test cases based on the tester's experience, knowledge and intuition.
<b>exploratory testing</b>	An informal test design technique where the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests. [After Bach]
<b>failure</b>	Deviation of the component or system from its expected delivery, service or result. [After Fenton]
<b>feature</b>	An attribute of a component or system specified or implied by requirements documentation (for example reliability, usability or design constraints). [After IEEE 1008]
<b>formal review</b>	A review characterized by documented procedures and requirements, e.g. inspection.
<b>Function Point Analysis (FPA)</b>	Method aiming to measure the size of the functionality of an information system. The measurement is independent of the technology. This measurement may be used as a basis for the measurement of productivity, the estimation of the needed resources, and project control.
<b>functional testing</b>	Testing based on an analysis of the specification of the functionality of a component or system. See also black box testing.
<b>generic goal</b>	A required model component that describes the characteristics that must be present to institutionalize the processes that implement a process area. [CMMI]
<b>generic practice</b>	An expected model component that is considered important in achieving the associated generic goal. The generic practices associated with a generic goal describe the activities that are expected to result in achievement of the generic goal and contribute to the institutionalization of the processes associated with a process area. [CMMI]
<b>heuristic evaluation</b>	A static usability test technique to determine the compliance of a user interface with recognized usability principles (the so-called "heuristics").
<b>higher level management</b>	The person or persons who provide the policy and overall guidance for the process, but to provide direct day-to-day monitoring and controlling of the process. Such persons belong to a level of management in the organization above the intermediate level responsible for the process and can be (but are not necessarily) senior managers. [CMMI]
<b>horizontal traceability</b>	The tracing of requirements for a test level through the layers of test documentation (e.g. test plan, test design specification, test case specification and test procedure specification or test script).



---

<b>impact analysis</b>	The assessment of change to the layers of development documentation, test documentation and components, in order to implement a given change to specified requirements.
<b>incident</b>	Any event occurring that requires investigation. [After IEEE 1008]
<b>incident logging</b>	Recording the details of any incident that occurred, e.g. during testing.
<b>incident management</b>	The process of recognizing, investigating, taking action and disposing of incidents. It involves logging incidents, classifying them and identifying the impact. [After IEEE 1044]
<b>incident management tool</b>	A tool that facilitates the recording and status tracking of incidents. They often have workflow-oriented facilities to track and control the allocation, correction and re-testing of incidents and provide reporting facilities. See also defect management tool.
<b>incident report</b>	A document reporting on any event that occurred, e.g. during the testing, which requires investigation. [After IEEE 829]
<b>independence of testing</b>	Separation of responsibilities, which encourages the accomplishment of objective testing. [After DO-178b]
<b>informal review</b>	A review not based on a formal (documented) procedure.
<b>input</b>	A variable (whether stored within a component or outside) that is read by a component.
<b>inspection</b>	A type of peer review that relies on visual examination of documents to detect defects, e.g. violations of development standards and non-conformance to higher level documentation. The most formal review technique and therefore always based on a documented procedure. [After IEEE 610, IEEE 1028] See also peer review.
<b>institutionalization</b>	The ingrained way of doing business that an organization follows routinely as part of its corporate culture.
<b>intake test</b>	A special instance of a smoke test to decide if the component or system is ready for detailed and further testing. An intake test is typically carried out at the start of the test execution phase. See also smoke test.
<b>integration</b>	The process of combining components or systems into larger assemblies.
<b>integration testing</b>	Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.
<b>level test plan</b>	A test plan that typically addresses one test level. See also test plan.
<b>managed process</b>	A performed process that is planned and executed in accordance with policy; employs skilled people having adequate resources to produce controlled outputs; involves relevant stakeholder; is monitored, controlled and reviewed; and is evaluated for adherence to its process description. [CMMI]
<b>management review</b>	A systematic evaluation of software acquisition, supply, development, operation, or maintenance process, performed by or on behalf of management that monitors progress, determines the status of plans and schedules, confirms requirements and their system allocation, or evaluates the effectiveness of management approaches to achieve fitness for purpose. [After IEEE 610, IEEE 1028]
<b>master test plan</b>	A test plan that typically addresses multiple test levels. See also test plan.
<b>maturity level</b>	Degree of process improvement across a predefined set of process areas in which all goals in the set are attained. [CMMI]
<b>milestone</b>	A point in time in a project at which defined (intermediate) deliverables and results should be ready.
<b>moderator</b>	The leader and main person responsible for an inspection or other review process.



<b>monitor</b>	A software tool or hardware device that runs concurrently with the component or system under test and supervises, records and/or analyses the behavior of the component or system. [After IEEE 610]
<b>non-functional testing</b>	Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, maintainability and portability.
<b>non-functional test design techniques</b>	Procedure to derive and/or select test cases for non-functional testing based on an analysis of the specification of a component or system without reference to its internal structure. See also black box test design technique.
<b>optimizing process</b>	A quantitatively managed process that is improved based on an understanding of the common causes of variation inherent in the process. The focus of an optimizing process is on continually improving the range of process performance through both incremental and innovative improvements.
<b>output</b>	A variable (whether stored within a component or outside) that is written by a component.
<b>pass/fail criteria</b>	Decision rules used to determine whether a test item (function) or feature has passed or failed a test. [IEEE 829]
<b>peer review</b>	A review of a software work product by colleagues of the producer of the product for the purpose of identifying defects and improvements. Examples are inspection, technical review and walkthrough.
<b>performance indicator</b>	A high level metric of effectiveness and/or efficiency used to guide and control progressive development, e.g. lead-time slip for software development. [CMMI]
<b>phase test plan</b>	A test plan that typically addresses one test phase. See also test plan.
<b>post condition</b>	Environmental and state conditions that must be fulfilled after the execution of a test or test procedure.
<b>precondition</b>	Environmental and state conditions that must be fulfilled before the component or system can be executed with a particular test or test procedure.
<b>pretest</b>	See intake test.
<b>priority</b>	The level of (business) importance assigned to an item, e.g. defect.
<b>process</b>	A set of interrelated activities, which transform inputs into outputs. [ISO 12207]
<b>process area</b>	A cluster of related practices in an area that, when implemented collectively, satisfy a set of goals considered important for making improvements in that area. [CMMI]
<b>process improvement</b>	A program of activities designed to improve the performance and maturity of the organization's processes, and the result of such a program. [CMMI]
<b>product risk</b>	A risk directly related to the test object. See also risk.
<b>project</b>	A project is a unique set of coordinated and controlled activities with start and finish dates undertaken to achieve an objective conforming to specific requirements, including the constraints of time, cost and resources. [ISO 9000]
<b>project risk</b>	A risk related to management and control of the (test) project, e.g. lack of staffing, strict deadlines, changing requirements, etc. See also risk.
<b>project test plan</b>	See master test plan.
<b>quality assurance</b>	Part of quality management focused on providing confidence that quality requirements will be fulfilled. [ISO 9000]
<b>quality attribute</b>	A feature or characteristic that affects an item's quality. [IEEE 610]
<b>quantitatively managed process</b>	A defined process that is controlled using statistical and other quantitative techniques. The product quality, service quality, and process-performance attributes are measured and controlled throughout the project. [CMMI]

<b>regression testing</b>	Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.
<b>release note</b>	A document identifying test items, their configuration, current status and other delivery information delivered by development to testing, and possibly other stakeholders, at the start of a test execution phase. [After IEEE 829]
<b>requirement</b>	A condition or capability needed by a user to solve a problem or achieve an objective that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. [After IEEE 610]
<b>requirements-based testing</b>	An approach to testing in which test cases are designed based on test objectives and test conditions derived from requirements, e.g. tests that exercise specific functions or probe non-functional attributes such as reliability or usability.
<b>requirements management tool</b>	A tool that supports the recording of requirements, requirements attributes (e.g. priority, knowledge responsible) and annotation, and facilitates traceability through layers of requirements and requirements change management. Some requirements management tools also provide facilities for static analysis, such as consistency checking and violations to pre-defined requirements rules.
<b>requirements phase</b>	The period of time in the software lifecycle during which the requirements for a software product are defined and documented. [IEEE 610]
<b>result</b>	The consequence/outcome of the execution of a test. It includes outputs to screens, changes to data, reports, and communication messages sent out. See also actual result, expected result.
<b>resumption criteria</b>	The testing activities that must be repeated when testing is re-started after a suspension. [After IEEE 829]
<b>re-testing</b>	Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions.
<b>review</b>	An evaluation of a product or project status to ascertain discrepancies from planned results and to recommend improvements. Examples include management review, informal review, technical review, inspection, and walkthrough. [After IEEE 1028]
<b>reviewer</b>	The person involved in the review that identifies and describes anomalies in the product or project under review. Reviewers can be chosen to represent different viewpoints and roles in the review process.
<b>review tool</b>	A tool that provides support to the review process. Typical features include review planning and tracking support, communication support, collaborative reviews and a repository for collecting and reporting of metrics.
<b>risk</b>	A factor that could result in future negative consequences; usually expressed as impact and likelihood.
<b>risk analysis</b>	The process of assessing identified risks to estimate their impact and probability of occurrence (likelihood).
<b>risk-based testing</b>	An approach to testing to reduce the level of product risks and inform stakeholders on their status, starting in the initial stages of a project. It involves the identification of product risks and their use in guiding the test process.
<b>risk control</b>	The process through which decisions are reached and protective measures are implemented for reducing risks to, or maintaining risks within, specified levels.
<b>risk identification</b>	The process of identifying risks using techniques such as brainstorming, checklists and failure history.
<b>risk level</b>	The importance of a risk as defined by its characteristics impact and likelihood. The level of risk can be used to determine the intensity of testing to be performed. A risk level can be expressed either qualitatively (e.g. high, medium, low) or quantitatively.

<b>risk management</b>	Systematic application of procedures and practices to the tasks of identifying, analyzing, prioritizing, and controlling risk.
<b>risk mitigation</b>	See risk control.
<b>risk type</b>	A specific category of risk related to the type of testing that can mitigate (control) that category. For example the risk of user-interactions being misunderstood can be mitigated by usability testing.
<b>root cause</b>	A source of a defect such that if it is removed, the occurrence of the defect type is decreased or removed. [CMMI]
<b>root cause analysis</b>	An analysis technique aimed at identifying the root causes of defects. By directing corrective measures at root causes, it is hoped that the likelihood of defect recurrence will be minimized.
<b>scribe</b>	The person who records each defect mentioned and any suggestions for process improvement during a review meeting, on a logging form. The scribe has to ensure that the logging form is readable and understandable.
<b>severity</b>	The degree of impact that a defect has on the development or operation of a component or system. [After IEEE 610]
<b>simulator</b>	A device, computer program or system used during testing, which behaves or operates like a given system when provided with a set of controlled inputs. [After IEEE 610, DO178b] See also emulator.
<b>smoke test</b>	A subset of all defined/planned test cases that cover the main functionality of a component or system, to ascertaining that the most crucial functions of a program work, but not bothering with finer details. A daily build and smoke test is among industry best practices. See also intake test.
<b>software lifecycle</b>	The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software lifecycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, retirement phase. Note these phases may overlap or be performed iteratively.
<b>specific goal</b>	A required model component that describes the unique characteristics that must be present to satisfy the process area. [CMMI]
<b>specific practice</b>	An expected model component that is considered important in achieving the associated specific goal. The specific practices describe the activities expected to result in achievement of the specific goals of a process area. [CMMI]
<b>specification</b>	A document that specifies, ideally in a complete, precise and verifiable manner, the requirements, design, behavior, or other characteristics of a component or system, and, often, the procedures for determining whether these provisions have been satisfied. [After IEEE 610]
<b>specified input</b>	An input for which the specification predicts a result.
<b>staged representation</b>	A model structure wherein attaining the goals of a set of process areas establishes a maturity level; each level builds a foundation for subsequent levels. [CMMI]
<b>state transition testing</b>	A black box test design technique in which test cases are designed to execute valid and invalid state transitions.
<b>statement coverage</b>	The percentage of executable statements that have been exercised by a test suite.
<b>statement testing</b>	A white box test design technique in which test cases are designed to execute statements.
<b>static analysis</b>	Analysis of software artifacts, e.g. requirements or code, carried out without execution of these software artifacts.

<b>static code analyzer</b>	A tool that carries out static code analysis. The tool checks source code, for certain properties such as conformance to coding standards, quality metrics or data flow anomalies.
<b>static testing</b>	Testing of a component or system at specification or implementation level without execution of that software, e.g. reviews or static code analysis.
<b>statistical testing</b>	A test design technique in which a model of the statistical distribution of the input is used to construct representative test cases.
<b>status accounting</b>	An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of the approved changes. [IEEE 610]
<b>stub</b>	A skeletal or special-purpose implementation of a software component, used to develop or test a component that calls or is otherwise dependent on it. It replaces a called component. [After IEEE 610]
<b>sub-practice</b>	An informative model component that provides guidance for interpreting and implementing a specific or generic practice. Sub-practices may be worded as if prescriptive, but are actually meant only to provide ideas that may be useful for process improvement. [CMMI]
<b>suspension criteria</b>	The criteria used to (temporarily) stop all or a portion of the testing activities on the test items. [After IEEE 829]
<b>syntax testing</b>	A black box test design technique in which test cases are designed based upon the definition of the input domain and/or output domain.
<b>system</b>	A collection of components organized to accomplish a specific function or set of functions. [IEEE 610]
<b>system integration testing</b>	Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).
<b>system testing</b>	The process of testing an integrated system to verify that it meets specified requirements. [Hetzel]
<b>technical review</b>	A peer group discussion activity that focuses on achieving consensus on the technical approach to be taken. [Gilb and Graham, IEEE 1028] See also peer review.
<b>test</b>	A set of one or more test cases. [IEEE 829]
<b>test approach</b>	The implementation of the test strategy for a specific project. It typically includes the decisions made that follow based on the (test) project's goal and the risk assessment carried out, starting points regarding the test process, the test design techniques to be applied, exit criteria and test types to be performed.
<b>test basis</b>	All documents from which the requirements of a component or system can be inferred. The documentation on which the test cases are based. If a document can be amended only by way of formal amendment procedure, then the test basis is called a frozen test basis. [After TMap]
<b>test case</b>	A set of input values, execution preconditions, expected results and execution post conditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement. [After IEEE 610]
<b>test case specification</b>	A document specifying a set of test cases (objective, inputs, test actions, expected results, and execution preconditions) for a test item. [After IEEE 829]
<b>test charter</b>	A statement of test objectives, and possibly test ideas about how to test. Test charters are used in exploratory testing. See also exploratory testing.

<b>test closure</b>	During the test closure phase of a test process data is collected from completed activities to consolidate experience, test ware, facts and numbers. The test closure phase consists of finalizing and archiving the test ware and evaluating the test process, including preparation of a test evaluation report. See also test process.
<b>test comparator</b>	A test tool to perform automated test comparison of actual results with expected results.
<b>test condition</b>	An item or event of a component or system that could be verified by one or more test cases, e.g. a function, transaction, feature, quality attribute, or structural element.
<b>test control</b>	A test management task that deals with developing and applying a set of corrective actions to get a test project on track when monitoring shows a deviation from what was planned. See also test management.
<b>test cycle</b>	Execution of the test process against a single identifiable release of the test object.
<b>test data</b>	Data that exists (for example, in a database) before a test is executed, and that affects or is affected by the component or system under test.
<b>test data preparation tool</b>	A type of test tool that enables data to be selected from existing databases or created, generated, manipulated and edited for use in testing.
<b>test design</b>	(1) See test design specification. (2) The process of transforming general testing objectives into tangible test conditions and test cases.
<b>test design specification</b>	A document specifying the test conditions (coverage items) for a test item, the detailed test approach and identifying the associated high level test cases. [After IEEE 829]
<b>test design technique</b>	Procedure used to derive and/or select test cases.
<b>test design tool</b>	A tool that supports the test design activity by generating test inputs from a specification that may be held in a CASE tool repository, e.g. requirements management tool, from specified test conditions held in the tool itself, or from code.
<b>test environment</b>	An environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test. [After IEEE 610]
<b>test estimation</b>	The calculated approximation of a result (e.g. effort spent, completion date, costs involved, number of test cases, etc.) which is usable even if input data may be incomplete, uncertain, or noisy.
<b>test evaluation report</b>	A document produced at the end of the test process summarizing all testing activities and results. It also contains an evaluation of the test process and lessons learned.
<b>test execution</b>	The process of running a test on the component or system under test, producing actual result(s).
<b>test execution phase</b>	The period of time in a software development lifecycle during which the components of a software product are executed, and the software product is evaluated to determine whether or not requirements have been satisfied. [IEEE 610]
<b>test execution schedule</b>	A scheme for the execution of test procedures. The test procedures are included in the test execution schedule in their context and in the order in which they are to be executed.
<b>test execution tool</b>	A type of test tool that is able to execute other software using an automated test script, e.g. capture/playback. [Fewster and Graham]
<b>test harness</b>	A test environment comprised of stubs and drivers needed to execute a test.
<b>test implementation</b>	The process of developing and prioritizing test procedures, creating test data and, optionally, preparing test harnesses and writing automated test scripts.



<b>test infrastructure</b>	The organizational artifacts needed to perform testing, consisting of test environments, test tools, office environment and procedures.
<b>test input</b>	The data received from an external source by the test object during test execution. The external source can be hardware, software or human.
<b>test item</b>	The individual element to be tested. There usually is one test object and many test items. See also test object.
<b>test level</b>	A group of test activities that are organized and managed together. A test level is linked to the responsibilities in a project. Examples of test levels are component test, integration test, system test and acceptance test. [After TMap]
<b>test log</b>	A chronological record of relevant details about the execution of tests. [IEEE 829]
<b>test logging</b>	The process of recording information about tests executed into a test log.
<b>test manager</b>	The person responsible for project management of testing activities and resources, and evaluation of a test object. The individual who directs, controls, administers, plans and regulates the evaluation of a test object.
<b>test management</b>	The planning, estimating, monitoring and control of test activities, typically carried out by a test manager.
<b>test management tool</b>	A tool that provides support to the test management and control part of a test process. It often has several capabilities, such as test ware management, scheduling of tests, the logging of results, progress tracking, incident management and test reporting.
<b>Test Maturity Model (TMM)</b>	A five level staged framework for test process improvement, related to the Capability Maturity Model (CMM), which describes the key elements of an effective test process.
<b>Test Maturity Model Integrated (TMMi)</b>	A five level staged framework for test process improvement, related to the Capability Maturity Model Integration (CMMI), which describes the key elements of an effective test process.
<b>test monitoring</b>	A test management task that deals with the activities related to periodically checking the status of a test project. Reports are prepared that compare the actuals to that which was planned. See also test management.
<b>test object</b>	The component or system to be tested. See also test item.
<b>test objective</b>	A reason or purpose for designing and executing a test.
<b>test performance indicator</b>	A high level metric of effectiveness and/or efficiency used to guide and control progressive test development, e.g. Defect Detection Percentage (DDP).
<b>test phase</b>	A distinct set of test activities collected into a manageable phase of a project, e.g. the execution activities of a test level. [After Gerrard]
<b>test plan</b>	A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process. [After IEEE 829]
<b>test planning</b>	The activity of establishing or updating a test plan.
<b>test policy</b>	A high level document describing the principles, approach and major objectives of the organization regarding testing.
<b>Test Point Analysis (TPA)</b>	A formula based test estimation method based on function point analysis. [TMap]
<b>test procedure specification</b>	A document specifying a sequence of actions for the execution of a test. Also known as test script or manual test script. [After IEEE 829]
<b>test process</b>	The fundamental test process comprises test planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities.

---

<b>Test Process Improvement (TPI)</b>	A continuous framework for test process improvement that describes the key elements of an effective test process, especially targeted at system testing and acceptance testing.
<b>test progress report</b>	A document summarizing testing activities and results, produced at regular intervals, to report progress of testing activities against a baseline (such as the original test plan) and to communicate risks and alternatives requiring a decision to management.
<b>test run</b>	Execution of a test on a specific version of the test object.
<b>test schedule</b>	A list of activities, tasks or events of the test process, identifying their intended start and finish dates and/or times, and interdependencies.
<b>test script</b>	Commonly used to refer to a test procedure specification, especially an automated one.
<b>test session</b>	An uninterrupted period of time spent in executing tests. In exploratory testing, each test session is focused on a charter, but testers can also explore new opportunities or issues during a session. The tester creates and executes test cases on the fly and records their progress. See also exploratory testing.
<b>test specification</b>	A document that consists of a test design specification, test case specification and/or test procedure specification.
<b>test strategy</b>	A high-level description of the test levels to be performed and the testing within those levels for an organization or program (one or more projects).
<b>test suite</b>	A set of several test cases for a component or system under test, where the post condition of one test is often used as the precondition for the next one.
<b>test summary report</b>	A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items against exit criteria. [After IEEE 829]
<b>test tool</b>	A software product that supports one or more test activities, such as planning and control, specification, building initial files and data, test execution and test analysis. [TMap]
<b>test type</b>	A group of test activities aimed at testing a component or system focused on a specific test objective, i.e. functional test, usability test, regression test etc. A test type may take place on one or more test levels or test phases. [After TMap]
<b>testability review</b>	A detailed check of the test basis to determine whether the test basis is at an adequate quality level to act as an input document for the test process. [After TMap]
<b>tester</b>	A skilled professional who is involved in the testing of a component or system.
<b>testing</b>	The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.
<b>test ware</b>	Artifacts produced during the test process required to plan, design, and execute tests, such as documentation, scripts, inputs, expected results, set-up and clear-up procedures, files, databases, environment, and any additional software or utilities used in testing. [After Fewster and Graham]
<b>traceability</b>	The ability to identify related items in documentation and software, such as requirements with associated tests. See also horizontal traceability, vertical traceability.
<b>unit test</b>	See component test.
<b>use case testing</b>	A black box test design technique in which test cases are designed to execute user scenarios.
<b>V-model</b>	A framework to describe the software development lifecycle activities from requirements specification to maintenance. The V-model illustrates how testing activities can be integrated into each phase of the software development lifecycle.



---

<b>validation</b>	Confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled. [ISO 9000]
<b>verification</b>	Confirmation by examination and through provision of objective evidence that specified requirements have been fulfilled. [ISO 9000]
<b>vertical traceability</b>	The tracing of requirements through the layers of development documentation to components.
<b>walkthrough</b>	A step-by-step presentation by the author of a document in order to gather information and to establish a common understanding of its content. [Freedman and Weinberg, IEEE 1028] See also peer review.
<b>white-box test design technique</b>	Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.
<b>white-box testing</b>	Testing based on an analysis of the internal structure of the component or system.
<b>Wide Band Delphi</b>	An expert based test estimation technique that aims at making an accurate estimation using the collective wisdom of the team members.

## 6 References

- [Bach] J. Bach (2004), Exploratory Testing, in: E. van Veenendaal, *The Testing Practitioner – 2nd edition*, UTN Publishing, ISBN 90-72194-65-9.
- [Beizer] B. Beizer (1990), *Software Testing Techniques*, van Nostrand Reinhold, ISBN 0-442-20672-0
- [BS7925-2] BS7925-2 (1998), *Standard for Software Component Testing*, British Standards Institution
- [Burnstein]. I. Burnstein (2002), *Practical Software Testing*, Springer Professional Computing, ISBN 0-387-95131-8
- [CMM] M. Paulk, C. Weber, B. Curtis and M.B. Chrissis (1995), *The Capability Maturity Model, Guidelines for Improving the Software Process*, Addison-Wesley, ISBN 0-201-54664-7
- [CMMI] M.B. Chrissis, M. Konrad and S. Shrum (2007), *CMMI Second Edition, Guidelines for Process Integration and Product Improvement*, Addison Wesley, ISBN 0-321-27967-0
- [DO-178b] DO-178 (1992). *Software Considerations in Airborne Systems and Equipment Certification, Requirements and Technical Concepts for Aviation (RTCA SC167)*,
- [Fenton] N. Fenton (1991), *Software Metrics: a Rigorous Approach*, Chapman & Hall, ISBN 0-53249-425-1
- [Fewster and Graham] M. Fewster and D. Graham (1999), *Software Test Automation, Effective use of test execution tools*, Addison-Wesley, ISBN 0-201-33140-3.
- [Freedman and Weinberg] D. Freedman and G. Weinberg (1990), *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, ISBN 0-932633-19-6.
- [Gelperin and Hetzel] D. Gelperin and B. Hetzel (1998), “*The Growth of Software Testing*”, in: CACM, Vol. 31, No. 6, 1988, pp. 687-695
- [Gerrard] P. Gerrard and N. Thompson (2002), *Risk-Based E-Business Testing*, Artech House Publishers, ISBN 1-58053-314-0.
- [Gilb and Graham] T. Gilb and D. Graham (1993), *Software Inspection*, Addison-Wesley, ISBN 0-201-63181-4.
- [Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), *Foundations of Software Testing*, Thomson Learning, ISBN 978-1-84480-355-2
- [Grochtmann] M. Grochtmann (1994), *Test Case Design Using Classification Trees*, in: Conference Proceedings STAR 1994.
- [Hetzel] W. Hetzel (1988), *The complete guide to software testing – 2nd edition*, QED Information Sciences, ISBN 0-89435-242-3.
- [IEEE 610] IEEE 610.12 (1990), *Standard Glossary for Software Engineering Terminology*, IEEE Standards Board
- [IEEE 829] IEEE 829 (1998), *Standard for Software Test Documentation*, IEEE Standards Board
- [IEEE 1008] IEEE 1008 (1993), *Standard for Software Unit Testing*, IEEE Standards Board
- [IEEE 1028] IEEE 1028 (1997), *Standard for Software Reviews and Audits*, IEEE Standards Board
- [IEEE 1044] IEEE 1044 (1993) *Standard Classification for Software Anomalies*, IEEE Standards Board
- [ISO 9000] ISO 9000 (2005) *Quality Management Systems – Fundamentals and Vocabulary*, International Organization of Standardization
- [ISO 9126] ISO/IEC 9126-1 (2001) *Software Engineering – Software Product Quality – Part 1: Quality characteristics and sub-characteristics*, International Organization of Standardization
- [ISO 12207] ISO/IEC 12207 (1995) *Information Technology – Software Lifecycle Processes*, International Organization of Standardization
- [ISTQB] ISTQB – E. van Veenendaal (ed.) (2007), *Standard Glossary of Terms Used in Software Testing, V2.0*, International Software Testing Qualifications Board

- [Koomen and Pol] T. Koomen and M. Pol (1999), *Test Process Improvement*, Addison-Wesley, ISBN 0-201-59624-5
- [TMap] M. Pol, R. Teunissen, E. van Veenendaal (2002), *Software Testing, A guide to the TMap Approach*, Addison Wesley, ISBN 0-201-745712.
- [Veenendaal] E. van Veenendaal (2004), *The Testing Practitioner – 2nd edition*, UTN Publishing, ISBN 90-72194-65-9.