A comparison of TMM and other Test Process Improvement Models

Author Ron Swinkels

Company Frits Philips Institute

Title A comparison of TMM and other Test Process Improvement Models.

Document Code 12-4-1-FP

Document Status Draft

Keywords Comparison, Test Process Improvement, TMM, MMAST, TAP, TCMM, TIM, TOM, TPI, TSM,

Abstract This report describes approaches (models) for Test Process Improvement and compares the approaches against the TMM.

Total of pages 51

Date 17-11-00

Introduction

This research is carried out as a part of the MB-TMM Project. The goal of the MB-TMM project is to develop a Test Process Improvement model. Other institutes and organizations have already developed various Test Process Improvement models. They all appeared recently, around the mid-nineties. The MB-TMM project uses the Testing Maturity Model (TMM), developed by Burnstein et al. (1996), as a point of departure. The TMM is supposed to be the most comprehensive Test Process Improvement model available yet. Unfortunately, only the outlines of the TMM test process improvement actions are available, while organizations desire more guidance. In addition, the TMM does not include a procedure to track/measure the improvements itself. These shortcomings, next to other shortcomings, must be tackled within the MB-TMM.

The idea behind this research is that we can learn from other Test Process Improvement models. They could contain ideas, which are important to consider when the MB-TMM must be developed. As a result, the purpose of this research is to describe and compare the most important Test Process Improvement models available and to give useful input for the development of the initial MB-TMM framework. Several models are studied on goals, structure, key process areas and its assessment procedure.

Next to this research, a SPI-research had been carried out. In this research, CMM, Spice and Bootstrap are discussed.

The Research Framework

Introduction

In this chapter, the framework for this Test Process Improvement-model research is discussed. First, the objectives of the research are explained. Then, all available Test Process Improvement models are presented. Not every model will be compared. Only those models will be selected for comparison, which satisfy several criteria. These criteria are discussed also in this chapter. At last, we will discuss how the selected research models are described and compared.

Objectives of the research

The outcomes of this research must serve as a source for developing the initial framework of the MB-TMM. Within the framework, the purpose, the maturity model structure, the key process areas, the assessment structure and eventually other elements will be determined. Therefore, this research must provide recommendations about these elements. It is supposed that research of the other Test Process Improvement models could provide useful input for the development of the MB-TMM. In addition, it is desired to get an overview of all available models at this moment.

So, the final objectives of this research are:

- Describe all available Test Process Improvement models at this moment.
- Compare selected models with TMM.
- Formulate recommendations for the development of an initial MB-TMM.

Available TPI-models

A small (informal) pre-investigation had taken place to gain an overview of the existing Test Process Improvement-models, which exist next to the TMM. The MB-TMM partners identified the next models:

- *The Maturity Model for Automated Software Testing (MMAST)*
- *The Testing Assessment Programme (TAP)*
- *The I.T.B.G. Testing Capability Maturity Model (TCMM)*
- *The Test Improvement Model (TIM)*
- *The Test Organization Maturity Model (TOM)*
- *The Test Process Improvement Model (TPI)*
- *The Testability Support Model (TSM)*

All the models are described below. A more elaborate description of the models can be found in the next chapters. Models that are not selected for comparison are more extensively described in the annexes.

The Maturity Model for Automated Software Testing (MMAST)

The Maturity Model for Automated Software Testing (MMAST), was developed by Mitchel H. Krause in 1994. The model is designed for manufacturers of computerized medical devices considering automating their software testing process.

The purpose of the MMAST is to determine the proper automation level that fits a manufacturing company of computerized medical devices the best. The model identifies a number of levels of automation, provides a guideline to determine at which level the organization has to develop and gives an overview of the expected costs for all levels of automation.

The model contains a:

- Maturity model
- Checklist
- Overview of associated costs

The Testing Assessment Programme (TAP)

Software Futures ltd and IE Testing Consultancy LTD designed the Testing Assessment Programme (TAP) in 1995. TAP is based on the Capability Maturity Model [Paulk et al. 1993a 1993b] and on Crosby's quality maturity scheme described in his book 'Quality is Free' (1979). According to the developers, the TAP includes people issues, which are neglected by the SEI model.

According to the developers of the model, the TAP helps to determine how a testing process can be improved. The TAP does not recommend the use of any particular technique, tool, organization style or standard.

The model contains a:

- Maturity model
- Questionnaire
- Scoring process

The I.T.B.G. Testing Capability Maturity Model (TCMM)

Susan M. Burgess and Rodger D. Drabick designed the Testing Capability Maturity Model (TCMM) in 1996. The TCMM is designed to provide an evaluation methodology to help determine organizations testing maturity.

The model consists of the following elements:

- Maturity model
- Questionnaire

The questionnaire is not discussed in this research, because it could not be obtained.

The Test Improvement Model (TIM)

The Testing Improvement Model was developed by Ericson, Subotic and Ursing. It is based on CMM and Gelperins TSM. TIM is intended to guide testing functions in their improvement work. According to the developers, TIM can be used in two ways:

- TIM is used to identify the current state of practice in key areas.
- TIM suggests ways in which strengths can be built upon and the weaknesses removed.

TIM consists of a:

- Maturity model
- Assessment Procedure

The Test Organization Maturity Model (TOM)

The Testing Organization Maturity Model has been developed at Systeme Evolutif. According to the developers, the existing maturity models (CMM) did not seem to address organizational problems well. The so-called 'remedy-based' models miss the point. Improvement methods that use remedy-based approaches are inadequate because they do not take existing problems, objectives and constraints into consideration.

According to the developers of TOM, major barriers to improved practices are organizational, not technical. Most of the difficulties in the implementation of improved practices are associated with changing management perceptions, overcoming people's natural resistance to change and implementing workable processes and management controls.

The purpose of TOM is to support, identify and prioritize the organizational bottlenecks and generate solutions for these problems.

The model consists of a:

- Questionnaire to identify and prioritize symptoms
- Improvement suggestions

The Test Process Improvement Model (TPI)

The TPI model has been developed by Koomen and Pol in 1997. One of the main reasons for developing the model was that testing is to be considered as an important, but difficult and uncontrolled process. The TPI model is developed to make improvement of the testing process easier. Koomen and Pol published a book about the TPI in 1999.

The TPI model offers a framework to determine the weak and strong areas of a testing process in an organization. In addition, the maturity of the process can be assessed. In addition, the TPI model can support determining improvement activities.

The TPI model contains the following parts:

- Maturity model
- Test maturity matrix
- Checklist
- Improvement suggestions

The Testability Support Model (TSM)

The Testability Support Model has been developed by David Gelperin in 1996. It is also known as the Testability Maturity Model. The TSM is designed to identify issues for improving testability. According to the information that could be acquired, the model only consists of a maturity model.

The Testability support Model has only three levels. TSM has six Key Support Areas, KSAs, analogous to the KPAs in the CMM:

- Software Engineering infrastructure.
- Project plans.
- Product information.
- Software design.
- Testware.
- Test environments.

Criteria to select models

The preceding part described all available models. Now, the models must be compared with the TMM. Of cause, the comparisons must produce useful proposals for the development of the MB-TMM. The Frits Philips Institute identified three criteria for selecting the models that must be satisfied in order to compare a model usefully.

1) The purpose of the model is to improve the whole test process
This research is carried out as a part of the MB-TMM project. The purpose of the MB-TMM project is to develop a test process improvement model considering all (important) test-related activities. Therefore, this research must only select the TPI-models that have the purpose to improve the whole test process.

2) The model must have a maturity structure
There are various approaches to improve a test process. There are also various test process improvement models available, each with its own scope and structure. An important characteristic of the TMM, the reference model of the MB-TMM project, is its maturity level structure. A maturity level is defined as a well-defined evolutionary plateau toward achieving a mature software process [Paulk et al 1993b]. Probably, the MB-TMM will contain a maturity level structure also. Therefore, this research must only select TPI-models, which contain a maturity level structure.

3) There is sufficient information available about the model to compare it usefully
The pre-investigation proved that it was hard to obtain TPI-model information. Sometimes, only a two-page description could be acquired. It is obvious that certain amounts of information are too little to compare the model usefully. Logically, only those models are described from which enough information could be acquired. Generally, this means that the information about the purpose, elements and contents of the Key Process Areas are available.

In the table on the next page, the models and their (non-) compliance to the criteria are shown. According to these criteria , only TIM and TPI could be selected. Therefore, the TIM and TPI focus on improving the whole test process and contain a TMM-like maturity model structure. In addition, there is enough model-information available of both models available to compare them with the TMM. The other models are included in the annexes.

	1) Improve whole process	2) Maturity Structure	3) Sufficient information	Selected to compared with TMM
MMAST	No	Yes	Yes	No
TAP	Yes	Yes	No	No
TCMM	Yes	Yes	No	No
TIM	Yes	Yes	Yes	**Yes**
TOM	No	No	No	No
TPI	Yes	Yes	Yes	**Yes**
TSM	Yes	Yes	No	No

Table 1: TPI-models mapped at the selection criteria.

Description and comparison of the models

This research describes and compares, if possible, the following elements of the TMM, TIM and the TPI model:

- Purpose of the model
- Elements where the model consists of
- Maturity Model Structure,
- Key Process Areas,
- Assessment Model/Procedure.

The TMM, TIM and TPI are discussed in the next chapters. We started with the TMM, because this is the reference model of the research. Then, the TIM and TPI are extensively described and compared. The author tried to analyze the not selected models (MMAST, TOM) also, because several MB-TMM partners thought that these models generate useful input for the development for the MB-TMM. It was not always possible to compare the models, because of lack of information. The analyses of the models are described in the annexes. TMM, TIM and TPI contain Key Process Areas[1] which are a cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability at that maturity level [Paulk et al. 1993B]. The identified Key Process Areas are compared in a similar way for every model:

- In which TMM-Goals (Key Process Area) do activities of the Key process Areas of the researched model recur?
- Describe, if possible, the most important similarities between the TMM Goals and the Key Process Area of the other model.
- Describe, if possible, the most important differences between the TMM Goals and the Key Process Area of the other model.

For every comparison between the model and TMM, conclusions are given. The conclusions relate at least to the main characteristic differences between the researched model and the reference model TMM. In addition, elements of the research model that could be used for MB-TMM are mentioned also.

The conclusions of all the comparisons lead to a final list of proposals for the development of the initial MB-TMM framework. These proposals are included in the last chapter of this report.

[1] Key Process Areas are also known as Goals or Key Areas

The Testing Maturity Model

Burnstein et al. from the Illinois Institute of Technology (IIT) have developed the Testing Maturity Model (TMM) in 1996. One of the main reasons for developing the TMM was that existing maturity models did not adequately address testing issues. In addition, the nature of a mature testing process was not defined well. The Testing Maturity Model has been developed to complement the software engineering oriented CMM. Compared with other testing maturity models, there is relatively much information available about TMM.

The Maturity levels and the Maturity Subgoals have been developed by Burnstein et al (1996). Suwannasart (1996) grounded the TMM. She researched literature extensively and defined Activities, Tasks and Responsibilities (practices) to the TMM.

The purpose of the TMM is to support assessment and improvement drives within an organization. It is to be used by:

- An internal assessment team to identify the current capability state.
- Upper management to initiate a testing improvement program.
- Software quality assurance engineers to develop and implement process improvement plans.
- Development teams to improve testing effectiveness.
- Users/Clients to define their role in the testing process.

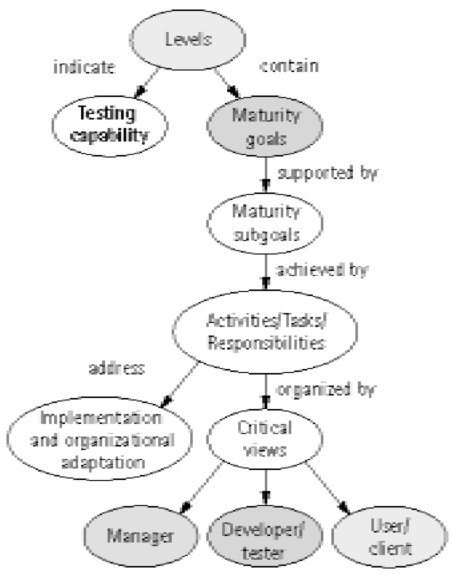
The model contains a:

- Maturity Model
- Assessment Model, containing an Assessment procedure, an Assessment instrument/questionnaire, team training and selection criteria.

Structure of the maturity model

The TMM structure contains several components, which are described below. The TMM-structure is shown in Figure 1.

FIGURE 1 The structure of the TMM



Maturity levels

Each maturity level represents a specific testing maturity in the evolution to a mature testing process. Movement to a higher level implies that lower-level practices are executed. There are five levels within TMM:

1. Initial
2. Phase Definition
3. Integration
4. Management and measurement
5. Optimization/defect prevention and quality control

Maturity Goals

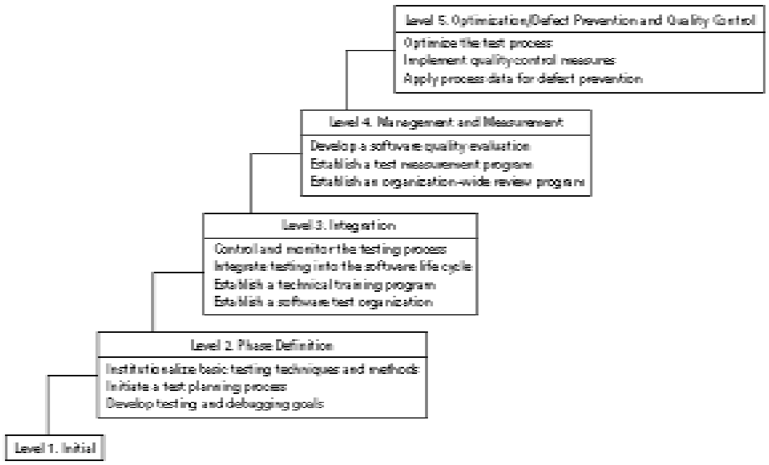
To achieve a certain maturity level, testing improvement goals have been identified that must be addressed. The maturity goals of TMM are equal to the Key Process

Areas of CMM. Goals are identified for each level, except for level 1. The goals are described in figure 2 on the next page and are further discussed in this chapter.

Maturity Subgoals.
For each maturity goal, subgoals that are more concrete are defined. For instance, a subgoal can concern the establishment of a test group, the distribution of a training programme, or the specification of items for review. The maturity goals of TMM are equal to the Key Practices of CMM.

Activities, Tasks and Responsibilities (ATRs)
The maturity subgoals are achieved through a set of Activities, Tasks and Responsibilities (ATRs). The ATRs address implementation and organizational adaptation issues at a specific level. Activities and Tasks are defined in terms of actions that must be performed to improve testing capability. Responsibility for these activities and tasks is assigned to the three key participants (critical views) in the testing process: managers, developers/testers, and users/clients. The ATRs of TMM are equal to the Activities of CMM had been developed by Suwannasart (1996). Examples of ATRs are:

FIGURE 2 TMM Maturity Goals by Level


level 5. Optimization/Defect Prevention and Quality Control
Optimize the test process
Implement quality control measures
Apply process data for defect prevention
Level 4. Management and Measurement
Develop a software quality evaluation
Establish a test measurement program
Establish an organization-wide review program
Level 3. Integration
Control and monitor the testing process
Integrate testing into the software life cycle
Establish a technical training program
Establish a software test organization
Level 2. Phase Definition
Institutionalize basic testing techniques and methods
Initiate a test planning process
Develop testing and debugging goals
Level 1. Initial

- Upper management provides adequate resources and funding from the committee on testing and debugging (manager view).
- Developers/testers work with managers to set testing and debugging goals (developers/testers view).
- The testing goals for each level of testing: unit, integration, system, and acceptance are set (developers/testers view).
- Users/clients meet with developers/testers to give their view of the nature of testing/debugging and policies. These goals and policies should represent their interests with respect to product quality (users/clients view).

Sources for development

Four sources served as inputs to TMM development:

The Capability Maturity Model
The CMM [Paulk et al 1993a, 1993b] is a process and evaluation model developed by the Software Engineering Instititute. It has been widely accepted and applied in the software industry. The CMM uses the concept of maturity models as a script for test process evaluation and improvement. The structure of the CMM consists of Maturity Levels, Key Process Areas, Key Practices and Activities.

The Gelperin and Hetzel's Testing model
The authors used this model as the foundation for defining the levels of TMM. The Gelperin and Hetzel Model (1988) describe phases and test goals from the 1950s through the 1990s. The initial period is debugging oriented, which describes the first level of TMM. The testing process has since then progressed to a prevention-oriented method, which reflects level 5 of the TMM.

Current industrial practices
The industrial practices survey of Durant (1993) described the best and worst testing environments at that time and allowed the TMM authors to extract benchmarks by which to evaluate and improve testing practices.

Beizer Progressive Phases of a Tester's Mental Model
The TMM authors have also included Beizers evolutionary model of the individual tester's thinking process. A mature testing organization is built on the skills, abilities and attitudes of the individual who works within it.

Description of the TMM Levels, Goals and Subgoals

In this part, each maturity level and its Goals will be described. An enumeration of the subgoals is also included.

Level 1: Initial

At this level, testing is a chaotic process. Tests are performed in a chaotic way after coding is done. The objective of testing is that software works. Software products are released without quality assurance. This maturity level does not contain any Goals.

Level 2: Phase definition

At this level, a testing phase is defined in the software life cycle. Testing is planned, is supported by basic testing techniques and tools, and is repeatable over all software projects. It is separated from debugging, the latter of which is difficult to plan. Maturity Goals include:

Develop testing and debugging goals
The organization must clearly distinguish between processes of testing and debugging. The separation of these two processes is essential for testing maturity growth, since they are different in goals and methods. Maturity subgoals include:

- The organization must form a committee(s) on testing and debugging with support and funding.
- The committee(s) must develop and record testing goals.
- The committee(s) must develop and debugging goals.
- The documented testing and debugging goals must be distributed to all project managers and developers.
- Testing goals must be reflected in the test plans.

Initiate a test planning process
Test planning involves stating objectives, analyzing risks, outlining strategies, and developing test design specifications and test cases. In addition, the test plan must address the allocation of

resources and the responsibilities for testing on the unit, integration, system and acceptance levels. Maturity subgoals include:

- An organization-wide test planning committee must be established with funding.
- A framework for organization-wide test planning policies must be established and supported by management.
- A test plan template must be developed, recorded and distributed to project managers.
- Project managers and developers must be trained to use the test plan template and develop the test plan.
- A procedure must be put in place that includes user-generated requirements as inputs to the test plan.
- Basic planning must be evaluated, recommended, and acquired. Management must support usage.

Institutionalize basic testing techniques and methods

To improve test process capability, basic testing techniques and methods must be applied across the organization. Policies for these techniques and basic tool support should be clearly specified. Examples of basic techniques and methods are black box and white-box testing strategies, use of the requirements validation matrix, and unit, integration, system and acceptance testing. Maturity subgoals supporting this Goal include:

- Management must institute a set of policies that ensures recommended techniques and methods that are consistently applied throughout the organization.
- An organization-wide test technology group must be formed to study, evaluate, and recommend a set of basic testing techniques and methods and recommend a set of simple tools to support them.

Level 3: Integration

This maturity level is essential for building quality into software products early in the software lifecycle. Testing at TMM level 3 is evolved into a set of well-defined activities that are integrated into all phases of the software life cycle. At this level management also supports the formation and training of a software test group.

Establish a software test organization

A software test organization is created to identify a group of people who are responsible for testing. The test group is responsible for test planning, test execution and recording, test-related standards, test metrics, the test database, test reuse, test tracking and evaluation. Maturity goals that support this goal include:

- An organization-wide test group must be established with leadership, support, and funding from upper management. Its functionality and position in the reporting must be defined.
- Roles and responsibilities must be defined for the test group.
- Well-trained and motivated staff members must be assigned to the test group.
- The test group must establish communication links, which asks user and client participation in testing activities to help collect, document, and incorporate user needs, concerns and requirements into the testing process.

Establish a technical training program
Testers must be properly trained so they can perform their jobs efficiently and effectively. At this level, the staff is trained in test planning, testing methods, standards, techniques, and tools. The training program also prepares the staff for the review process, instructing review leaders and instituting channels for user participation in the testing and review processes.
Training includes in-house courses, self-study, mentoring programs, and support for attendance at academic institutions. Maturity subgoals include:

- Management must establish an organizational training program, providing funding and support.
- The technical training committee must develop and distribute an organizational training policy.
- Training goals and plans must be developed with input from project managers.
- An in-house training group must be established, with tools, facilities and materials in place.

Integrate testing into the software lifecycle
Test planning must be initiated early in the life cycle. The testers and developers use a variation of the V-model. User input to the testing process is soliced through established channels for several of the testing phases. Maturity subgoals include:

- The test phase must be partitioned into subphases that can be integrated into the software lifecycle following a written organizational policy and reviewed with management.
- An institutionalized version of the V-model, based on defined testing subphases, must be developed and adopted.
- Resources must be provided to support the integration of testing activity into the software life cycle.
- Standard definitions must be developed for test-related work products and its quality must be measured.
- A procedure must be established that allows testers to work with developers to facilitate testing activity.

Control and monitor the testing process
The controlling and monitoring activities provide visibility and ensure the testing process proceeds according to plan. Support for controlling and monitoring comes from standards for test products, test milestones and test metrics that can be used to evaluate test progress and test effectiveness. Maturity goals include:

- The organization must develop mechanisms and policies to control and monitor the testing process.
- A set of test process-related measurements must be defined, recorded and distributed.
- A set of corrective actions and contingency plans must be developed, recorded, and documented for use when testing deviates significantly from what is planned.

Level 4: Management and Measurement

The principal focus areas at this level are to broaden the definition of a testing activity and accurately the measurements. The definition of a testing activity is expanded to include reviews, inspections and walk-through at all phases of the life cycle. Software work products as well as test-related work products such as test-plans, test designs, and test procedures are all reviewed. This expanded definition of testing covers activities typically categorized as verification and validation activities. The primary goal of this broadened set of testing operations is to uncover them as early as possible.

Establish an organization-wide review program

At TMM level 3, an organization integrates testing activities into the lifecycle. At level 4, this integration is augmented by the establishment of a review program. Reviews are conducted at all phases of the lifecycle to identify, catalog and remove defects from software work products and to test work products early and effectively. Maturity goals include:

- Upper management must develop review policies, support the review process, and take the responsibility for integrating them in the organizational culture.
- The test group and the software quality assurance group must develop and document goals, follow-up procedures, and recording mechanisms for reviews throughout the software lifecycle.
- The above bodies must specify items for review.
- Personnel must be trained so that they understand and follow proper review policies, practices and procedures.

Establish a test measurement program

A test measurement program is essential to accurately evaluate the test process quality, the productivity of the personnel and the progress of test process improvement. Measurement data for every test lifecycle phase must be specified. Measurements include those related to test progress, test costs, data on errors and defects, and product measures such as software reliability. Maturity subgoals that support this goal include:

- Organization-wide test measurement policies and goals must be defined.
- A test measurement plan must be developed with mechanisms for data collection, analysis, and application.
- Action plans that apply measurement results to test process improvements must be developed and documented.

Software quality evaluation

Software quality evaluation involves defining measurable quality attributes and defining measurable quality attributes to evaluate software work products. Quality is tied to testing process adequacy because a mature testing process must lead to software that is at least correct, reliable, usable, maintainable, portable and secure. Maturity subgoals include:

- Management and the testing software quality assurance groups must define quality related policies, quality goals, and quality attributes for software products.
- The organization must develop, document, and support procedures and policies for evaluating software quality.
- The testing process must be structured, measured and evaluated to ensure that the quality goals can be achieved.

Level 5: Optimization/Defect prevention and quality control

Organizations at this level test to ensure that the software satisfies its specification and the organization can establish a certain level of confidence in its reliability. There are tests to detect and prevent faults in the life cycle. The testing process is now planned, organized, staffed, controlled and directed. Management provides leadership and motivation and supports the infrastructure necessary for the continual improvement of product and process quality.

Application of process data for defect prevention

Organizations at this level record defects, analyze defect patterns and identify root causes of errors. Action plans are developed, actions are taken to prevent defect recurrence and there is a mechanism to track action progress. Defect prevention is applied across all projects and across the organization. Maturity subgoals include:

- The organization must develop, document, and support procedures and policies for defect prevention.
- A defect prevention team must be established with management support.
- Defects injected or removed must be identified and recorded during each lifecycle phase.
- A causal analysis mechanism must be established to identify the root causes of defects.
- Action plans must be developed through the interaction of managers, developers and testers to prevent identified defects from recurring; these plans must be tracked.

Quality control

At this level organizations use statistical sampling, measurements of confidence levels, trustworthiness and reliability to drive the testing process. The cost to achieve quality goals is measured relative to the cost of not testing for quantitave quality goals. Subgoals include:

- The organization must develop, document, and support procedures for quality control.
- The software test group and the SQA group must establish goals for quality products such as product unit defectiveness, confidence levels and trustworthiness.
- Test managers must incorporate these quality goals into test plans.
- The test group must be trained in statistical methods.
- User input must be collected for usage modeling.

Test process optimization

At the highest level of the TMM, the testing process is subject to continuous improvement across projects and across the organization. The test process is quantified and can be fine-tuned so that capability growth is an ongoing process. Optimizing the testing process involve: identifying testing practices that need to be improved, implementing the improvements, tracking improvements progress and continuous evaluation of new test-related tools and technologies for adaptation. Maturity subgoals include:

- The organization must develop, document, and support procedures and policies for test process optimization.
- A test process improvement group must be established to monitor the testing process and identify areas for improvement.
- A mechanism must be in place to evaluate new tools and technologies that may improve the capability and maturity of the testing process.
- The effectiveness of the testing process must be continually evaluated, and decisions on when to stop testing must be related to quality goals and made in a measurable and optimal manner.

Assessment model of TMM

The TMM Assessment Model (TMM-AM) supports self-assessment of the testing process. It uses the TMM as its reference model. The TMM-AM is not intended to be used for certification of the testing process by an external body. The ISO, SW-CMM and SPICE Assessment Models were used to guide development of the TMM-AM [Paulk et al. 1995,1993a,1993b; ISO 1995; Zubrow et al. 1994].

The TMM-AM has three major components:

- The assessment procedure,
- The assessment instrument/questionnaire,
- Team training and selection criteria.

The Assessment Procedure

The TMM-AM consists of a series of steps that guide an assessment team in carrying out a testing process self-assessment. The principal goals for the TMM assessment procedure are:

- To support the development of a test process profile and the determination of a TMM level.
- To guide the organization in developing action plans for test process improvement.
- To ensure the assessment is executed with efficient use of the organizations' resources.
- To guide the assessment team in collecting, organizing, and analyzing the assessment data.

A summary of the steps in the assessment procedure is given beneath:

- Preparation. This step includes selecting and training the assessment team, choosing the team leader(s), developing the assessment plan, selecting the projects, and preparing the organizational units participating in the assessment.
- Conducting the assessment. In this step, the team collects and records assessment information from interviews, presentations, questionnaires, and relevant documents. The TMM traceability matrix, as described later, can be used to check data accuracy, consistency, and objectivity.
- Reporting the assessment outputs. The TMM-AM outputs include a process profile, a TMM level, and the assessment record.
- Analyzing the assessment outputs. The assessment team, along with management and software quality engineers, now use the assessment outputs to identify and prioritize improvement goals.
- Action planning. An action-planning team develops plans that focus on high-priority improvements identified in the previous step.
- Implementing improvement. After the action plans have been developed and approved, they are applied to selected pilot projects. The pilot projects need to be monitored and tracked to ensure task progress and goal achievement.

TMM questionnaire

This instrument supports the collection and analysis of information from an assessment, maintain a record of results and provide information for post mortem analysis.

The TMM questionnaire consists of eight parts:

1. Instructions for use

2. Respondent background
3. Organizational background
4. Maturity goal and subgoal questions
5. Testing tool use questions
6. Testing trends questions
7. Recommendations for questionnaire improvement
8. A glossary of testing terms

The questionnaire in not the sole source of input for determining the TMM level and generating testing assessment results. Information should be also collected from interviews, presentation and inspection of relevant documents. A shortcoming of the TMM assessment model is, that it is only available as a web-form and that no scoring

Assessment Training and Team Selection Criteria

The authors of TMM have adopted SPICE guidelines for selecting and preparing an effective assessment team [ISO 1995]. A trained assessment team is made up of members from within the organization. Assessment team members should understand assessment goals, have the proper knowledge, experience, have strong communication skills, and should be committed to test process improvement [Homyen 1998].

Tools for assessment support

To support an assessment team, the authors have developed tools to ensure that the assessments are performed in a consistent, repeatable manner to reduce assessor subjectivity and to ensure the validity, usability, and comparability of the assessment results. A description of the tools is given below:

1. Team training data recording template. This template allows the team leader to record and validate team-training data.
2. Traceability matrix. The traceability matrix, in conjunction with the assessment team training procedure, the team data recording template, and the traceability matrix review is introduced to address the issue of interrater agreement and general assessment reliability [El Emam et al. 1996]. The matrix, which is completed as data are collected, allows the assessors to identify sources of data, crosscheck the consistency and correctness of the data, and resolve any data-related issues.

The Testing Improvement Model

The Testing Improvement Model was developed by Ericson, Subotic and Ursing. It is based on CMM and Gelperins TSM. The model was developed because the developers thought that focus on test process improvement was necessary. Testing accounts for 35 to 50 percent of the software project process costs [Bender 1996], but testing state of practice is not as good as it ought to be.

TIM is intended to guide testing functions in their improvement work. According to the developers, TIM can be used in two ways:

- TIM is used to identify the current state of practice in key areas.
- TIM suggests ways in which strengths can be built upon and the weaknesses removed.

TIM consists of a:

- Maturity model
- An Assessment Procedure

Structure of the maturity model

The structure of TIM consists of a five-level ladder and key areas. The initial level 0 is a non-compliance level. Level 1-4 have an overall goal associated with subgoals. TIM currently has five Key Areas (KAs). The KAs of TIM are similar to the KSAs in TSM, as KAs and KSAs span multiple levels. Hence, KAs differ from CMM's KPAs that are confined to one level.

The names of the level reflect the goal of the level:

0. Initial
1. Baselining
2. Cost-effectiveness
3. Risk-lowering
4. Optimizing

Baselining
First, a baseline has to be established. The baseline is to have a testing function i.e. a group of people that are dedicated to testing. They have roles, responsibilities, and documented, standardized ways of conducting work.
Strategies for achieving the overall goals include:

- Analysis and classification of product problems.
- Documentation of standards, methods, and policies.

Cost effectiveness
When a baseline has been established, efforts can be focused on becoming cost-effective. Efforts are made to increase the efficiency of product problem detection. Under the assumption that schedules and resources remain the same, more product problems have to be detected. The strategies used include computer aid, training, re-use, and early detection. Good intra-team communication is facilitated by co-location. That is, the team should be at the same place at the same time, preferably working on the same project. Strategies for achieving the overall goals include:

- Early detection.

- Computer aid of testing tasks.
- Training.
- Re-use.

Risk-lowering

On the risk lowering level, risks are the focus. Risk management is about being prepared for, and acting on, undesired effects; that is, there should be a plan that addresses how to deal with the undesired effects. In order to reach the risk lowering level, the testing function takes a step towards, and initiates communication and co-operation with development. Ideally, the testing function should participate in all deployment areas. The management support should, at least result in tester influence on schedule and resource deployment. At the risk lowering level metrics are put to use. Metrics are used for determining fulfillment of test objectives, cost/benefits analysis of test activities in order to justify expenses, and detecting fault-prone test objects.

Strategies for achieving the overall goals include:

- Early involvement.
- Expense justification.
- Analysis of product and process problems.
- Measurement of product, process, and recourses.
- Risk analysis and risk management.
- Communication with all parties of projects.

Optimizing

A testing function at the optimizing level is capable of assuring the right quality. The optimizing testing function is likely to have a positive attitude towards continuous improvement. Management gives its full support since testing can show the business value of the improvement efforts. Knowledge and understanding is acquired through experimentation and modeling.

Strategies for achieving the overall goals:

- Knowledge and understanding through experimentation and modeling.
- Continuous improvement.
- Root cause analysis.
- Balancing of testing against product needs and organization’s quality policy.
- Co-operation with all parties of projects, in all phases of development.

Description of Key Areas

The Key Areas of TIM are similar to the KSAs in TSM, as Key Areas and KSAs span multiple levels. Hence, Key Areas differ from CMM's Key Process Areas that are confined to one level. The five Key Areas of TIM are:

- Organization
- Planning and tracking
- Testware
- Test cases
- Reviews

Each KA consists of related activities distributed over the four levels. The activities of all the key areas are described below:

	Organization
Baselining	Testing faction is organized according to documenting standard. There exists a test leader and a core test team.
Cost-effectiveness	Testing function is co-located. Key people work full-time on one project and resources are not shared. Independent testing functions. Team structure is based on resource stocktaking and project needs. Responsibilities are clearly defined, documented, and understood. Test group has a dedicated project area, and proper tools and utensils. Training to meet organizational needs.
Risk-lowering	Test sensitive resource allocation and task planning is performed at all levels. Testing function strives after support from all levels of management. Tester involvement in all development phases, and development involvement in testing. Personnel rotate between development and testing.
Optimizing	Testers are part of multidisciplinary teams. Continuous organizational improvement. Groups exist for process improvement.

	Planning and tracking
Baseling	Basic planning is performed. Entry and exit criteria are defined for all testing levels. Test results are documented and duly processed and distributed. Plans are developed following standards and deviations are documented and motivated. Planning is performed according to documented policies. Plans change with requirements. Changes to the test plan are made according to a documented policy.
Cost-effectiveness	Planning and tracking is computer aided. Generic plans are used. The choice of test methods and test levels is balanced against test objects and test objectives. Progress is measured and compared against plans. Planning is performed by a trained tester or planner. Planning is performed in an evolutionary manner. Resource prediction is done according to documented policy.
Risk lowering	Risk analysis is performed and influences planning and resource deployment. Affected parties approve plans, including test objectives. Fulfillment of overall test objectives is monitored. Planning is performed by an experienced tester/planner. Plan change with factual circumstances.
Optimizing	Models are made of costs/resources, risks, etc. Planning and tracking activities are continuously improved based on analysis metrics and experiences. Post-mortems are performed and results duly distributed.

	Test cases
Baselining	Test cases are revised when requirements change. Test cases are based on requirements. Test cases are designed and documented according to a documented policy, before execution. Test cases are executed according to documented instructions.
Cost-effectiveness	Test cases are designed using documented techniques. Testability aspects influence requirements and design. Test cases are recorded and re-used. Test cases are organized with respect to testing levels, requirements, test object, objectives, etc.
Risk-lowering	Test cases are ranked and selected according to criticality. Test cases are designed in response to risk areas.
Optimizing	Test case design techniques are measured, reviewed, and improved.

Testware

Testware is the actual testing procedures that are run, the support software, the data sets that are used to run the tests and the supporting documentation. The testware key area includes configuration management of testware and the use of testware and tools.

	Testware
Baselining	Problem reporting is computerized. File systems are used for configuration management.
Cost-effectiveness	Computer aided problem tracking, code tracking, and test execution. Computer aided test case result gathering, recording, processing, and reporting. Coverage analysis tools are used. Hardware and software systems are limited. Databases are used for configuration management of testware. Test tools are evaluated before they are bought and put into action. Static analysis tools are used, e.g. to prepare code for reviews.
Risk-lowering	Computer aided risk analysis. Testware and software is under effective configuration management. Computer aided regression testing. Only mature technology is used.
Optimizing	Computer aided result checking and evaluation. Computer aided testware design. Test environment is integrated. Experimentation with tools.

Reviews

TIM refers to reviews in general, embracing all techniques involving reading or visual inspection of software documents. Examples of review techniques are inspection, technical review, walkthrough, peer review, and code reading.

	Reviews
Baselining	Requirements are reviewed. Review leaders and management are given training in review techniques. Training and support is bought in for at least the first project. Standardized, documented review techniques are used.
Cost-effectiveness	All review staff is provided training review techniques. Checklists, scenarios etc. are customized and used. Design and code documents are reviewed. The organization has some review techniques to choose from.
Risk lowering	Testware and test cases are reviewed. Review techniques are evaluated. Review processes, products, and resources are measured.
Optimizing	Review techniques are continuously improved. Selection of team and technique is based on facts.

The Assessment procedure of TIM

The assessment procedure consists of two parts. The actual assessment, where the status of the organization is determined, is followed by the development of an improvement plan. The assessment part comprises the following three steps:

1. The models and the assessment procedure are explained to individuals in the organization in question. Representatives from all levels of the organization should participate in the briefing.
2. The actual assessment is performed as interviews, with strategically selected representatives from the organization. The model is used in conjunction with a questionnaire in order to determine where to focus the attention. It is unclear if the model itself contains a standard questionnaire.
3. The results of the interviews are analyzed and presented as a 'maturity profile' where the organization’s 'score' for each Key Area is given.

The development of an improvement plan includes:

- Solution identification. Given the maturity profile, TIM suggests strategies that can be used to improve the organization. The strategies form a solution space.
- Solution analysis. The solution space is combined with the needs and visions of the organization. Considering what strategies are relevant, important and cost-effective for the organization reduces solution space. The possible strategies are discussed and ranked, according to capability, resources, timing and management opinion.
- Presentation. The proposed improvement tasks are combined with the results of the assessment.

An alternative, lightweight assessment procedure can be used when an organization feels uneasy about undergoing a formal assessment. During the course of one day, the assessor and representatives from the organization reason about their operation. The downside is that no plan for improvement is developed.

Note: an assessment document (questionnaire) could not be acquired.

Experience

In the article of Ericson, three TIM cases are discussed. Ericson stated that the testing process improved after using TIM at one case. The process improvement of the organization in the second case did not work out well, because there was no management commitment that supported improvements. The third organization used TIM, but no second assessment has been carried out. Therefore, process improvement could not be measured.

Comparison with TMM

Purpose

Both models have the same purpose. TIM as well as TMM identify the current state of practice in key process areas and contain suggestions to improve the testing process.

Elements

Both models contain a maturity model, but the structure of the maturity models differs significantly. TMM has the most comprehensive Assessment Model. TIM contains a smaller assessment procedure, and no training and selection criteria. We do not know if TIM contains a questionnaire.

Structure

Both TIM and TMM have a five level structure. However, the Key Areas of TIM span all the maturity levels, whereas TMM-Goals are confined to one level. Both models have subgoals. TMM discusses how these subgoals are achieved through a set of activities, tasks and responsibilities. However, TIM does not have identified how subgoals are achieved.

Key Areas

Organization
Corresponding TMM Goals
Establish a software organization (3)

Similarities
The importance of establishing a test organization is recognized in both TIM and TMM. Responsibilities need to be defined. Management commitment is also indispensable. In both models, the testing organization is established at a higher level. Also testing involvement in the complete software life cycle is recognized in both models.

Differences
TIM specifies the details of the testing organization, for instance: testing function must be co-located, there needs to be job-rotation between testing and development. TMM emphasizes the communication aspect of the testing organization to other groups.
TIM pays little attention to training, whereas training in TMM is a separate KPA (Establish a technical training program).

Planning and tracking
Corresponding TMM Goals
Initiate a planning Process (2)

Control and monitor the testing process (3)

Similarities
In both models planning and controlling/tracking are fundamental elements for Test Process Improvement. Plans must be developed against a standard/template. The test plan must be distributed. In both models, planning tools are mentioned.

Differences
TIM assigns planning and tracking at the first improvement level, whereas TMM distinguishes two planning goals: Initiate a planning process at level 2 and Control and monitor the testing process at level 3. In addition, an important difference is that TMM identifies corrective actions when testing deviates from planning, whereas TIM does not.

Test cases
Corresponding TMM Goals
Initiate a test planning process (2).

Similarities
TMM only mentions test cases as a responsibility for the test group.

Differences
According to TMM, defining test cases are a part of the test planning process. Test cases in TMM are by no means extensively discussed. In TIM, test cases seem to be a primary aspect of the testing process.

Testware (and tools)
Corresponding TMM Goals
Institutionalize basic testing techniques and methods (2)

Similarities
Both models prescribe that problems and test execution must be tracked and that testtools must be evaluated before they are brought into action.

Differences
TIM emphasizes the utilization of a computer system. TIM also prescribes what types of tools (coverage analysis tools, static tests) must be used.

Review
Corresponding TMM Goals
Establish an organization-wide review program (4)

Similarities
Both models claim that reviewing is an important aspect of the testing process. In addition, there needs to be a standardization of review techniques. People need to be trained for review techniques.

Differences

TMM emphasizes the organizational aspects of training techniques; issues such as development of review policies and procedures, embedding reviews in the organizational culture are discussed. TIM describes what needs to be reviewed, for instance: requirements, design documents and code documents.

The Assessment model

The contents of the TIM assessment procedure are also mentioned in TMM. Both models describe the phases: representatives-selection, interviews, analyzing results, determining maturity level and improvement selection.

However, TMM discusses more aspects of an assessment. For instance: TMM suggests that an assessment team has to be selected and trained, that assessment outputs are a process profile, a TMM level and an assessment record. The TMM has a structured Assessment questionnaire available, whereas it is not known if TIM contains one. The TMM assessment model is more extensive than the TIM assessment procedure.

Conclusion

The most striking difference in the structure of both models is that Key Areas of TIM span all the maturity levels, whereas TMM-Goals are confined to one level. One of the fundamental questions arise from this conclusion: does only the test organization evolve according to the maturity principle, or do the aspects in the model also, and must this be taken into account in the development of a MB-TMM?

It looks like TIM views the testing process at a lower level. TIM discusses some test-related elements, like testcases and testware, at a more detailed level than the TMM. However, TIM discusses less testing-aspects, especially at the more mature levels. For instance, Defect prevention and Quality control are not discussed in TIM.

The TMM looks at the testing process at a high, broader, organizational view. TMM emphasizes more at the development/specification of groups, committees, policies and responsibilities, and discussing the contents. TMM also covers more aspects than TIM does, although we will not state that TMM is exhaustive.

However, it cannot be posed that the models contradict with each other. Every model is a selection of important areas and these areas are described with a certain depth and from a certain point of view. In short, we can pose that TIM is a model, which describes a number areas more detailed, whereas TMM describes more areas (not all) less detailed.

The assessment models of TIM and TMM look similar. Roughly, they contain the same phases. However, it can be stated that the TMM assessment model is more comprehensive and detailed.

The Test Process Improvement model

The TPI has been developed by Koomen and Pol in 1997. One of the main reasons for developing the model was that testing is to be considered as an important, but difficult and uncontrolled process. The TPI could be a guideline to make improvement of the testing process easier. Koomen and Pol published a book about TPI in 1999.

The TPI model offers a framework to determine the weak and strong areas of a testing process in an organization. In addition, the maturity of the process can be assessed. In addition, the TPI model can support determining improvement activities.
The TPI model contains the following parts

- A maturity model
- The test maturity matrix
- A Checklist
- Improvement suggestions

Structure of the maturity model

Maturity levels

The Test Process Improvement Model has 3 basic maturity levels and 14 scales. Each level consists of a number of scales. These scales indicate which key areas need to be improved, as shown in the test maturity matrix discussed further in this chapter.

Controlled
The fist five scales are meant to get the testing process under control. The process must have a phased character and must be in accordance with a defined testing strategy. Specification techniques are used for testing, defects are reported and communicated. The testware and testing administration is managed.

Efficient
The sixth to tenth scale is meant to establish an efficient testing process. The testing process is automated, integrated and anchored in the development organization.

Optimizing
The last three scales are characterized by an optimizing nature. Continuous improvement to optimize the organization is part of the regular methods of the organization.

Description of Key Areas

There are 20 Key Areas in the TPI model. Most of the areas are related to the improvement of system and acceptation tests. For more mature testing processes, there are also areas related to white-box testing types.

There are different levels for each Key Area identified. The lowest level for a Key Area is A and the highest possible level is mostly C. The number of levels differs per area. To be assigned to a certain level, the Key Area activities need to satisfy all the checkpoints of the key-area level. The Key Areas are:

Test strategy
The test strategy must be focused on detecting the most important defects as soon as possible, at minimal costs. The strategy defines which requirements and quality risks are discovered by what tests.

Life-cycle model
Within the test process, a number of phases can be defined, such as planning, preparation, specification, execution and completion. Activities have to be executed for each phase. For each activity the goal, input, process, output, dependencies, etcetera has to be established.

Moment of Involvement
Normally, tests are executed after software development realization. Nevertheless, the testing process must be started much earlier, in order to remove defects earlier and cheaper.

Estimation and Planning
The testing planning and estimation indicate which activities have to be carried out when and how many resources/people are needed.

Test Specification Techniques
A Test Specification Technique is defined as a standardized way of deriving testcases from output information. Applying these techniques gives insight into the quality and depth of the tests and increases the reusability of the test.

Static Test Techniques
Work product checking without running programs and evaluate measures, which must lead to a certain quality level, is called a static test. Checklists are useful tools.

Metrics
Metrics are quantified observations of product/process characteristics. Metrics are important to track the progress of the test process and the quality of the system. Metrics are used to control the testing process, to ground testing-advices and to compare different systems and/or processes.

Test tools
Test tools are automated aids for testing. Using testing tools can have significant benefits, like shorter lead times and more testing flexibility.

Test Environment
The test execution takes place in a so-called test environment, which consists of the following components: procedures, hardware, software, means of communication, facilities for building and using databases and files The environment has a significant influence on the quality, lead-time and costs of the testing process. Important aspects are: responsibilities, administration, on-time and sufficient availability, representativeness and flexibility.

Office Environment

The test staff needs rooms, desks, chairs, PCs, and so on in order to work efficiently. A good and timely office environment has a positive influence on the motivation.

Commitment and Motivation

The commitment and motivation of the persons involved in testing are important prerequisites for a smoothly running test process. The persons involved are not only the testers, but also, for example, the project management and the line management personnel.

Testing Functions and Training

Within a testing process, it is important to have a correct composition of a test team. A right mix of different disciplines, functions, knowledge and skills is required. Trainings are needed to acquire the needed mix.

Scope of Methodology

For each test process, a certain methodology or working method is used, comprising activities, procedures, regulations, techniques, etc. The aim is that the organization uses a methodology which is sufficiently generic to be applicable in every situation, but which contains enough detail so that it is not necessary to rethink the same items again each time.

Communication

Within a testing process, communication is an important aspect. These communication forms are important for a smoothly running test process, not only to create good conditions and to optimize the test strategy, but also to communicate about the progress and the quality. This Key area handles the internal communication of the test team.

Reporting

Reporting should be aimed at giving well-founded advice to the customer concerning the product and even the system development process.

Defect Management

Good management should be able to track the life cycle of a defect and supports the analysis of quality trends in the detected defects. Such analysis is used, for example, to give well-founded quality advice.

Testware Management

The products of testing (testware) should be maintainable and reusable and so they must be managed. Besides the products of testing themselves, such as test plans, specifications, databases and files, it is important that the products of previous processes such as functional design and realization are managed well, because the test process can be disrupted if the wrong program versions, that are delivered.

Test Process Management

To control every activity and every process, four steps are essential: planning, execution, maintaining and adjusting. These steps need to be identified for every activity/process.

Evaluation
Evaluation means inspecting intermediate products such as the requirements and the functional design. The significant benefit is that defects can be detected earlier in the process. This saves time and money. In addition, recovery costs drop when tests are executed in an early stage.

Low level testing
The developers almost exclusively carry out the low-level tests. Well-known low-level tests are the unit test and the integration test. Low-level testing is efficient, because it requires little communication and because often the finder is both the error producer as well as the one who corrects the defect.

The Test Maturity Matrix

When the testing process is assessed, the level of each key area is identified. After the assessment, improvement activities can be established with the test maturity matrix. It is not sensible to raise all key area level B. For instance, for introducing Metrics at level A, Reports and Defect Administration need to be at level A. Therefore, key areas are mutually dependent and therefore need to be improved in a certain order. This order is expressed in the three-category/fourteen-scale Testing Maturity Matrix. After determining the process capability, the Matrix can be used to determine improvement actions.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Test strategy		A					B				C		D	
Life-cycle model		A			B									
Moment of involvement			A				B				C		D	
Estimating and planning				A							B			
Test specification tech.		A		B										
Static techniques					A		B							
Metrics						A			B			C		D
Test Tools					A			B			C			
Test Environment				A				B						C
Office Environment				A										
Commitment and Motivation		A				B						C		
Testing functions and training				A			B			C				
Scope of methodology					A						B			C
Communication			A		B							C		
Reporting		A			B		C					D		
Defect management		A				B		C						
Testware management			A			B				C				D
Testing process management		A		B								C		
Evaluation							A			B				
Low-level testing					A		B		C					
	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Figure 3: the test maturity matrix

The Checklist

To determine the level of a key area, the model uses checkpoints as a measuring instrument. The checklist contains a number of checkpoints for each level for each key area. All the checkpoints of a certain level need to be satisfied in order to be categorized in that level. These checkpoints are cumulative, so for level B, the level A checkpoints of the area also needs to be satisfied.

Examples of checkpoints are:

- The test group consists at least of a test manager and a couple of testers
- One or more forms or informal test specifications techniques are used, suitable for the needed depths of tests
- The testing activity starts together with or earlier then the demands are formulated

Improvement suggestions

Another tool is the list of improvement suggestions. These suggestions are meant as hints and tips to improve the Key Area, not compulsory steps. Every level of every key area has a number of improvement suggestions.

Comparison with TMM

Purpose

Both models have the same purpose. The TPI model as well as the TMM identify the current state of practice in key areas and are meant to improve the testing process.

Elements

Both models contain a maturity model, but the structure of the maturity models differs significantly. The assessment models are quite similar. Instead of a questionnaire (TMM), the TPI model uses checkpoints.

Structure of the maturity model

The TPI model specifies three levels, instead of five levels used by the TMM. In contrast with the TMM, these levels are not grounded on a theoretical basis. The TPI model contains a lot more key areas than the TMM (20 versus 13). These Key Areas are not confined to one level.

Key Areas

Test Strategy
Corresponding TMM Goals
Initiate a test planning process (2).
Integrate testing in the Software-lifecycle (3)

Similarities/Differences
A test strategy is part of the planning process within TMM. The TPI model describes the test strategy more extensively, by testing type (black box, white box, review)

Life-cycle model
Corresponding TMM Goals
Initiate a test planning process (2).

Similarities/Differences
Both models state that tests must be planned. The TPI model has more clearly distinguished phases than the TMM; Planning, Specification, Execution and Completion. The phase 'Specification' within the TMM looks part of the 'Planning' phase.

Moment of involvement
Corresponding TMM Goals
Integrate testing into the life cycle (3).

Similarities/Differences
Both models recognize that testing must be executed early in the software development life cycle. Within TPI, the moment of involvement must be earlier as the Key Area level increases. TMM refers to the V-Model. Testing methods, either verification or validation have to be developed for each stage in the V-model. According to TMM, also attention has to be given to cooperation between testers and developers.

Estimating and planning
Corresponding TMM Goals
Institutionalize basic testing techniques and methods (2).

Similarities/Differences
The planning activity is discussed in both models, whereas TMM mentions more aspects; stating objectives, analyzing risks, outlining strategies, etcetera. Within the Estimating and Planning Key Area, TPI focuses on allocation of resources (people).

Test specification techniques
Corresponding TMM Goals
Initiate a test planning process (2).

Similarities/Differences
Both models state that test cases need to be specified. TMM mentions that test planning policies must be identified. TPI describes the testing specification more detailed.

Static test techniques
Corresponding TMM Goals
Institutionalize basic testing techniques and methods (2)
Establish an organization-wide review program (4)

Similarities/Differences
Both models discuss the use of techniques. TMM does not mention static techniques separately.

Metrics
Corresponding TMM Goals
Control and monitor the test process (3), Establish a measurement program (4)

Similarities/Differences
Within the TPI, the metrics are prescribed in detail for the project/process/system/organization. The TMM does not, as it focuses on the monitoring/controlling process and the corrective actions that have to be taken. Metrics is only one aspect of the process.

Test tools
Corresponding TMM Goals
None.

Similarities/Differences
TMM does not explicitely handle test tools whereas TPI does. Test tools are 'spread out'over various TMM-KPAs.

Test environment
Corresponding TMM Goals
None.

Similarities/Differences
The TMM does not handle the testing environment.

Office environment
Corresponding TMM Goals
None.

Similarities/Differences
The TMM does not handle the office environment

Commitment and Motivation
Corresponding TMM Goals
Most Goals of the TMM.

Similarities/Differences
Management support recurs in most Goals of the TMM. According to the TMM, management support is a condition to realize every Goal. Within the TPI model, management support grows from assigning budget and time to test engineering accepted.

Testing functions and training
Corresponding TMM Goals
Establish a software testing organization (3), Establish a technical training program (3), Quality control (5).

Similarities/Differences
Both models describe that tasks and responsibilities need to be assigned. The TPI describes some functions extensively. Within the TMM, the training part is more extensively described than in the TPI model; there must be a separate training in-house training group, training goals and plans must be developed. The TPI only states that the personnel must have sufficient education to perform their tasks.

Scope of methodology

Corresponding TMM Goals

Institutionalize basic testing techniques and methods (2), Initiate a test planning process (2)

Similarities/Differences

Both models state that methods, like procedures and techniques need to be applied. However, within the TMM procedures have to be immediately defined on organizational level, the TPI allows methods defined on project level (Key area level A).

Communication

Corresponding TMM Goals

Establish a software test organization (3), Initiate a test planning process (2)

Similarities/Differences

According to the TMM, the test organization has to establish communication links. The TPI describes the communication procedures much more detailed. Communication as defined by TPI is considered to be part of the planning issues.

Reporting

Corresponding TMM Goals

None.

Similarities/Differences

The TMM does not handle Reporting.

Defect Management

Corresponding TMM Goals

None.

Similarities/Differences

The TMM does not handle Defect Management.

Testware Management

Corresponding TMM Goals

None.

Similarities/Differences

The TMM does not handle Testware Management.

Test process Management

Corresponding TMM Goals

Initiate a test planning process (2), Control and monitor test process (3)

Similarities/Differences
Most activities within the models are similar; designing a test plan, controlling progress. The TPI distinguishes corrective actions on the testing department and organizational level, whereas the TMM does not.

Evaluation
Corresponding TMM Goals
Establish an organization wide review program (4)

Similarities/Differences
Both models discuss the reviews at a higher level of maturity. Reviews must be executed early in the development cycle. The TPI model points at the use of risk analysis to determine which critical work products need to be reviewed (thoroughly).

Low level testing
Corresponding TMM Goals
Institutionalize basic testing techniques and methods (2)

Similarities/Differences
The TPI emphasizes the activities for each white box-phase. The TMM states that (white box) testing types need to be studied, recommended, applied and checked on use.

Assessment Procedure

The TPI identifies two useful tools: the test validation matrix and the checklist. Both are needed to assess on which maturity level an organization is. The checklist consists of more questions than the TMM questionnaire. TPI does does deal with the preperation of the assessment, although not that elaborate.

The TMM contains a procedure, a questionnaire a team training data-recording template, and a traceability matrix.

Conclusion

TPI and TMM are probably the most comprehensive models discussed in this research. However, there seems to be important differences between the models.
The maturity levels of the TPI are not based scientific research. The TMM used the Gelperin and Hetzel's model as a foundation. However, all improvement actions generally take place in the same order. The Key Areas span multiple (sub)levels, whereas the TMM Goals are confined to one level.

TMM does not handle the following TPI Key Areas:

- Test environment
- Office Environment
- Reporting
- Defect Management
- Testware Management

It has to be studied/discussed if these Key Areas must be implemented in the MB-TMM. The TPI uses other assessment tools than the TMM, but the purpose of the assessment itself is equal. The checklist of the TPI is more detailed than the TMM questionnaire.

Proposals for the MB-TMM

Introduction

In this final chapter, the TMM itself is compared with the TIM and TPI to get an insight in the quality of TMM. Then, a final list of proposals included for the development of the elements of the MB-TMM framework.

The TMM re-viewed

In comparison with the TIM and TPI, see also Table 2[2], the TMM of Burnstein is worked out rather well. The maturity as well as the assessment model seems to be quite complete. In addition, the TMM is based on research literature. Several dissertations [Suwannasart 1996, Grom] about the TMM were published. Therefore, the TMM seems one of the best 'model of depart' for the development of a test process improvement model such as the MB-TMM.

	TMM (reference)	**TIM**	**TPI**
Type Model	Maturity	Maturity	Maturity
Year of development	1996	1996	1997
Approach	Conceptual	Practical	Practical
Maturity model foundation	Gelperin and Hetzel	None	None
Number of levels	5	5	14
Number of KPAs	13	5	20
KPA type	One level	Multiple level	Multiple level
Coverage of test activities	+	+-	++
Assessment type	Questionnaire	Questionnaire	Checklist
Assessment elements	Assessment Procedure, Questionnaire, Training and Team Selection criteria	Assessment guideline Questionnaire	Assessment guideline Checklist Test Maturity Matrix
Assessment foundation	CMM, ISO, SPICE	Practical experience	Practical experience
Information about model	Articles Dissertations	Articles (a few)	Articles Book
Ideas for MB-TMM	N.A.	Different types of assessments	Improvement suggestions Checklist Test, environment Office environment Reporting Defect management Testware management
Remarks	Dissertations difficult to obtain.	Not much information available.	

Table 2: an overview of characteristics of the TMM, TIM and TPI.

Unfortunately, the dissertation of Suwannasart was not easily available. Some of the MB-TMM project partners thought that there were no ATR descriptions, whereas they were described in Suwannasarts dissertation! This is a pity. The author thinks that (ease of) availability of information is one of the key factors to have a model that is widely used, by different organizations. Therefore,

[2] A table of the charactaristics of all models is included in Annex F.

when the MB-TMM maturity model, assessment model and other tools are developed, they have to be disseminated well. Important information must be easily available on the web.

Key Process Areas of the TIM and TPI span multiple levels, whereas TMM KPAs do not. The multiple level concept could be an idea to divide TMM Goals, which contain many activities, over multiple levels.

A few activities did not seem to be addressed through TMM.

- Test environment (handled by TPI)
- Office Environment (handled by TPI)
- Reporting (handled by TPI)
- Defect Management (handled by TPI)
- Testware Management (handled by TPI)

The TPI contains improvement suggestions for each Key Area. Improvement suggestions are best practices that can be used when an organization wants to implement the activities, which the Key Area prescribes. The idea of guidelines could also be included in the MB-TMM.

The MMAST (see annex A) gives an overview of the associated costs when improvement activities are carried out. TMM does not. This could be an idea for development later in the MB-TMM project.

The developers of the TAP (see annex B) stated that CMM did not address people issues well. Because TMM is based on CMM, the adoption of people issues in the MB-TMM must be researched. This was also proposed by a test-expert from the MB-TMM project.

A list of proposals for the MB-TMM

In this part, several proposals for development of the MB-TMM are given. Of course, all these proposals need to be studied. This list is certainly not complete. Other researches can show other proposals. The proposals for MB-TMM are:

- CMM(I)/IEEE terminology is used.
- MB-TMM is based on TMM.
- Multiple level Key Areas for comprehensive combined activities (for instance: develop a testing organization).
- Inclusion of Key Process Area *Test environment* (handled by TPI).
- Inclusion of Key Process Area *Office Environment* (handled by TPI).
- Inclusion of Key Process Area *Reporting* (handled by TPI).
- Inclusion of Key Process Area *Defect Management* (handled by TPI).
- Inclusion of Key Process Area *Testware Management* (handled by TPI).
- Inclusion of *People issues* (according to TAP).
- The MB-TMM assessment model is based upon TMM/SPICE/ISO.
- An overview of the associated costs could be an idea later in the development process.
- A dissemination plan of MB-TMM has to be written.
- Web based access for MB-TMM (like SEI CMMI).

References

Beizer, B., Software system testing techniquesm second edition. New York: Van Norstrand Reinhold, 1990.

Bender, R., Proposed Software Evaluation and Test KPA, revision no. 4, Available September 10, 1996, http://www.softtest.com/pages/kpabai.htm.

Burnstein, I., Homyen, A., Questionnaire for the testing Maturity Model version 1.1, Illinois Institute of technology, April 1998, http://www.cs.iit.edu/~tmm.

Burnstein, I., Homyen, A., Suwannasart., T., Saxena, G., Grom, R., A Testing Maturity Model for Software Test Process Assessment and Improvement, Software Quality Professional, 1999, http://sqp.asq.org/vol1_issue4/sqp_v1i4_burnstein.html.

Burnstein, I., Suwannasart., T., Carlson, C.R., Developing a Testing Maturity Model, Part II, 1996, http://www.stsc.hill.af.mil/Crosstalk/1996/sep/developi.htm.

Burgess, S, M; Drabick R, D, The I.T.B.G.Testing Capability Maturity Model, 1996, http://www.iti-inc.com/TCMM.htm.

Crosby, P. D., Quality is free: the art of making quality certain, London, McGraw-Hill, 1979

Daich, G. T., Frazer, T., George, L., Ginsberg, M., Payne, R., Russell, G., Stream, J., and Tucker, J. White paper for recommending improvements to the verification and validation (V&V) activities in the capability maturity model, 1994.

Durant, J., Software testing practices survey report (TR5-93). Software Practices Research Center, 1993

El Emam, K., Goldenson, D., Briand, L., Marshall. Interrater agreement in SPICE-based Assessments: Some preliminary reports, Proceedings of the Fourth International Conference on the Software Process. Los Alamitos, Calif.: IEEE Computer Society Press1996.

Ericson, T., Subotic, A., Ursing, S., TIM-A Test Improvement Model, Software Testing Verification and Reliability 7(4), pp. 229-246.

Gelperin, D., How to support better software testing, Application development Trends 3(5), 42-8, 1996a.

Gelperin, D., A Testability Support Model, Presented at the fifth International Conference On Software Testing, Analysis & Review, May 13-17, Orlando, Florida, 1996b

Gelperin, D., Software Testability Support Model, October 1996c, http://homepage2.rconnect.com/glewwe/TwinSPIN/archive-96oct.html.

Gelperin, D., Hetzel, B., The growth of software testing. Communications of the Association of Computing Machinery 31, no6: 687-695, 1988.

Grom, R., Report on a TMM assessment support tool. Chicago: Illinois Institute of Technology.

Homyen, A., An assessment model to determine test process maturity., Ph D. thesis, Illinois Institute of Technology, 1998.

Houston, F., Software development and Quality Assurance: FDA Expectations, Proceedings of the 1992 HIMA Conference, HIMA Publication 93-5, Washington, DC, Health Industry Manufacturers Association, pp 4351, 1993.

Humphrey, W. S., Managing the Software Process, MA, Addison-Wesley, 1989.

International Organization for Standardization. ISO/IEC Software process assessment working draft-Part 3: Rating processes, version 1.0, Part 5: Construction, selection of use of assessment instruments and tools, version 1.0, Part 7: Guide for use in process improvement, version 1.0. Geneva, Switzerland: International Organization for Standardization.

I.T.B.G., writers unknown, The I.T.B.G. Testing Capability Maturity Model, http://www.iti.inc.com/TCMM.htm.

Koomen, T., Pol, M., Test Process Improvement, A practical step-by-step guide to structured testing, Addison-Wesley, 1999.

Krause, M. E. A Maturity Model for Automated Software Testing, Medical Device & Diagnostic Industry Magazine, December 1994, http://www.devicelink.com/mddi/archive/94/12/014.html.

Olsen, K., Staal Vinje, P., Using the Testing Maturity Model in practical test-planning and post-evaluation, presented at EuroSTAR98.

Paulk, M., Weber, C., Curtis., B., Chrissis. M., The capability maturity model guideline for improving the software process. Reading, Mass.: Addison-Wesley, 1995.

Paulk, M., Weber, C., Curtis., B., Chrissis, M., Capability maturity model, version 1.1., IEEE Software 10, no4 18-27, 1993a.

Paulk, M., Weber, C., Garcia, S., Chrissis, M., Bush, B., Key Practices of the capability maturity model, version 1.1 (CMU/SEI-93-TR-25). Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1993b.

Rakatin, S. R., The Economics of Software Process Improvement, presented to the 1994 Medical Device Software Conference sponsored by the Health Industry Manufacturers Association, Washington, DC, May 1994.

Suwannasart, T., Towards the development of a testing maturity model. Ph.D thesis, Illinois Institute of technology, 1996.

Zubrow, D., Hayes, W., Siegel, J., Goldenson, D., Maturity questionnaire (CMU/SEI-94-SR-7) Pittsburgh: Software Engineering Institute,1994.

Annex A: The Maturity Model for Automated Software Testing

The Maturity Model for Automated Software Testing (MMAST), was developed by Mitchel H. Krause in 1994. The model is designed for manufacturers of computerized medical devices considering automating their software testing process.

The purpose of the model is to determine the proper automation level that fits a manufacturing company of computerized medical devices the best. The model identifies a number of levels of automation, provides a guideline to determine at which level the organization has to develop and finally gives an overview of the expected costs at the level of automation.

The model contains:

- A Maturity model
- A checklist
- An overview of associated costs

Structure of MMAST: maturity levels only

The MMAST contains four levels, and is based on Houston (1993) and Rakatins (1994) adapted software testing maturity model originally developed by Watts S. Humphrey (1989). The maturity levels represent to what extent a testing process can be automated. The four levels are:

1. Accidental Automation:
2. Beginning Automation
3. Intentional Automation
4. Advanced Automation

Level 1: Accidental Automation
The first level is characterized by ad hoc, individualistic, chaotic attempts to get the job done. Important information is not documented. Test plans are sketchy. Either products are delayed or testing become a cursory, poorly documented exercise. This level has been designated Accidental Automation because the use of any automated tools or techniques comes about almost as if by accident and is not supported by process, planning, or management functions.

Level 2: Beginning Automation
Documentation is beginning to appear in the form of software requirements specifications (SRSs) and requirements specifications (TSRs). Automated testing tools are used, but have limited capabilities and lose their usefulness quickly as a product changes.

Level 3: Intentional Automation
At the third level, automated testing becomes both well defined and well managed. The TRSs and the test scripts themselves proceed logically from the SRSs and design documents. Because the test team is now part of the development process, these documents are written before the product is delivered for testing.

Level 4: Advanced Automation
The highest testing maturity level is a practiced and perfected version of level 3 with one major addition: post release defect tracking. The software test team is an integral part of product

development, and testers and developers work together to build a product that will meet test requirements.

There are no Key Process Areas, Key Practices or Key Activities described in MMAST.

Assessment: a checklist of issues

An assessment, to determine on which maturity level an organization is, is not presented in the MMAST. However, the MMAST includes a checklist that helps to identify at which level the testing process has to be automated. The checklist consists of four questions:

1. How large are your software projects?
2. How complex is your product?
3. What financial risk does your product pose for the company?
4. What risk does your product pose for the patient and operator?

Every question has a four-scale score rating. The needed automation level equals to the highest score at any question.

In addition, the MMAST describes associated improvement costs of process, people and tools

Comparison with TMM

Purpose

The purpose of the MASST differs completely from the TMM. The MMAST focuses on what level the testing process of a medical device manufacturer have to be automated. The TMM has a different purpose, it focuses on software test organizations in general, and helps to determine organizations test process maturity and describes the activities needed to achieve a higher level.

Elements

The elements of the MMAST are rather different from the TMM. The maturity model of the MMAST is not as elaborate as the TMM. The MMAST does not contain an assessment procedure.

A MMAST-checklist, which helps to identify a needed level of maturity, is not included in the TMM. In addition, the TMM does not contain an identification of associated costs.

Structure

The structure of the MMAST differs clearly from the TMM. Maturity levels are described and show roughly the same maturity stages. Though, the MMAST contains four instead of five maturity level. In addition, it does not contain Key Process Areas, or (Key) Activities.

Key areas

Important activities, such as establishing test requirements and defect tracking, can be derived from the maturity level description. However, MMAST does certainly not identify 'real' specific Key Areas. The model does not describe what goals have to be achieved for a certain testing maturity level.

Assessment

There is no assessment procedure in the MMAST, which identifies the actual performance level of an organization.

Conclusion

The MMAST and the TMM are far from similar. The key issue of the MASST is that a certain elements (products, regulations, risks) influence the decision on what level the testing process of a medical device manufacturer has to be automated.
The TMM has a different intention; it focuses on software test organization in general, helps to assess organizations' test process maturity level and describes the activities needed to achieve a higher level.

The MMAST does not describe any Key Practices Areas and there is no assessment procedure. In short: the MMAST defines a goal, but does not describe how it must be reached. However, this does not mean that the model is totally useless. The fundamental question of the MMAST is: to what extent needs an organization needs to be improved, is not addressed by TMM. MMAST also gives an overview of the associated costs when improvement activities are carried out. These elements are worth mentioning.

Annex B: The Testing Assessment Programme

Software Futures ltd and IE Testing Consultancy LTD designed the Testing Assessment Programme (TAP) in 1995. TAP is based on the Capability Maturity Model [Paulk et al. 1993a 1993b] and on Crosby's quality maturity scheme described in his book 'Quality is Free' (1979). The TAP helps to determine how a testing process can be improved. The TAP does not recommend the use of any particular technique, tool, organization style or standard. According to the developers, the TAP includes people issues, which are neglected by the SEI model.

The model contains a:

- Maturity model
- Questionnaire
- Scoring process

Maturity model structure and Assessment of TAP

The Testing Assessment Programme has five levels of maturity:

1. Initial Ad hoc, Chaotic
2. Repeatable Intuitive
3. Defined Qualitative
4. Managed Quantitative
5. Optimizing Continuous Improvement

The TAP assesses maturity in four Key Areas:

- Goals - why are we doing the testing?
- People - who is affected and do they understand the Goals?
- Management - how will the processes be controlled?
- Techniques - what techniques are needed to make the processes work?

First, testing goals must be clearly defined so that all efforts are focused on these goals. To implement testing work to meet the goals, people with the right skills, capabilities and motivation must be chosen. To manage the test activities as they proceed through the life cycle effective management methods must be in place. Finally, methods, techniques and tools should be used as appropriate to support any of the other areas. Within each category, there are a number of key activities, which the TAP assesses. The objective is to improve evenly across all areas. For example achieving a high maturity for techniques will not be productive if goals are overlooked or immature.

Comparison with TMM

The description above is the only public information available and is considered insufficient to compare the TAP with the TMM usefully. However, the following note is worth mentioning: according to the developers of TAP: CMM neglects people issues. The MB-TMM project management must determine if this must be studied for the TMM. It could be that the MB-TMM needs to be expanded with a separate People element.

Annex C: The Testing Capability Maturity Model

Susan M. Burgess and Rodger D. Drabick designed the Testing Capability Maturity Model (TCMM) in 1996. The TCMM is designed to provide an evaluation methodology to help determine organizations testing maturity.

The model consists of the following elements:

- Maturity model
- Questionnaire

The questionnaire is not discussed in this research, because it could not be obtained.

The TCMM was customized for the Federal Aviation Administration and used to assess the Testing Maturity of the ACT-200 program at their William J. Hughes (Atlantic City) Technical Center in September 1996. A final report with an accompanying set of testing process improvement recommendations was provided to the FAA in October 1996 in completion of the contract.

Structure of the maturity model

The model has five maturity levels (like the SEI CMM) and an established set of Testing Key Process Areas that provide a basis for an assessment questionnaire.

The Testing Capability Maturity Model has five levels of maturity:

1. Initial
2. Repeatable
3. Defined
4. Managed
5. Optimizing

The model contains Key Process Areas for each testing capability maturity level. In addition, a set of questions was developed for each KPA so that organizational maturity could be evaluated for each KPA. This in turn provided the basis for a quantitative measurement against each maturity level. The TCMM questionnaire is supported by an Excel spreadsheet for evaluating the responses and graphically displaying the summary results.

Comparison with TMM

The description above is the only public information available and is considered insufficient to compare the TCMM with TMM usefully.

Annex D: The Test Organization Maturity Model

Introduction

The Testing Organization Maturity Model has been developed at Systeme Evolutif. According to the developers, the existing maturity models (CMM) did not seem to address organizational problems well. The so-called 'remedy-based' models miss the point. Improvement methods that use remedy-based approaches are inadequate because they do not take existing problems, objectives and constraints into consideration.

According to the developers of TOM, major barriers to improved practices are organizational, not technical. Most of the difficulties in the implementation of improved practices are associated with changing management perceptions, overcoming people's natural resistance to change and implementing workable processes and management controls. To address the management problem, a mix of improvements is most likely to be required.

The purpose of TOM is to support identify and prioritize the organizational bottlenecks and generate solutions for these problems.

The model consists of a:

- Questionnaire to identify and prioritize symptoms
- Improvement suggestions

The improvement suggestions could not be obtained. Therefore, only the questionnaire is discussed.

The Questionnaire

The questionnaire is consists of twenty questions/symptom descriptions and focus on organizational rather than technical issues. A score has to be given to each question, reflecting to what extent the organization has tackled the problem. In addition, symptoms must be prioritized; reflecting which symptom the organization wants to tackle first. Furthermore, objectives of the organization assessed are taken into consideration and are prioritized. In most cases, management or practitioners can provide answers.

The scoring system is relatively simple. All scores and priorities are rated from one to five and can be added up. The improvement model is a scoring/weighting calculation that prioritizes potential improvements, based on the questionnaire scores and priorities.

When completed, the questionnaire can be used to calculate a TOM level. The completed questionnaire must be sent to Systeme Evolutif and will be processed in a database that has a built-in improvement model. The model has a library of 83 potential testing improvements. Based on the assessment data entered, the model generates up to seventy prioritized improvement suggestions.

Experience

About 120 organizations, ranging from different industries, have used TOM to improve their testing process.

Comparison with TMM

Purpose

The purpose of TOM is different from TMM. TOM generates solutions for a specific organizational problem(s). TMM is a guidance to improve the whole testing process.

Elements

Because the purposes of the models are different, the elements are not the same. TOM does not contain a maturity model and an assessment model, whereas TMM does not contain a symptom-oriented questionnaire. Though both TOM and TMM contain improvement suggestions, presented in a different way. TOM presents the improvement suggestions after the questionnaire has been sent to Systeme Evolutif. The maturity model of TMM are the improvement suggestions itself.

Structure of TOM

TOM does not contain a maturity model. It does not describe key areas, and does not use an improvement model. TOM tries to tackle problems, not to improve testing organizations as a whole.

Identified Improvement Activities

TOM describes different bottlenecks/symptoms. The model does suggest improvement activities after assessing the questionnaire, so that organizational bottlenecks can be identified. Information about the improvement activities is only available after an assessment. Therefore, it is difficult to compare improvement activities with Goals of the TMM.

Assessment model

It could be stated that TOM has an assessment model; the questionnaire. However, it does not look like the TMM Assessment model. Its purposes are completely different. The purpose of the TMM assessment model is to determine the maturity level of an organization, by determining which organizational elements have been satisfied. The TOM questionnaire helps to identify prescribed symptoms/organizational bottlenecks and gives a guidance to prioritize these problems.

Conclusion

The TOM differs fundamentally from the TMM. TOM does not use a maturity model and so, does not identify Key Areas. Instead, it tries to define the organizational bottlenecks (symptoms) of an organization and suggests improvements the specific problem. In addition, TOM focuses mainly on organizational test planning problems, Questions are mostly about: test planning. Therefore, TOM is not a guide for broad process improvement.

It would be prejudiced to say that the model is bad or useless. Not every organization wants to implement an extensive CMM/TMM for maturity growth. For some organizations, TOM could be an alternative to tackle (improve) a single/few bottlenecks.

Annex E: Testability Support Model

The Testability Support Model has been developed by David Gelperin in 1996. It is also known as the Testability Maturity Model. The TSM is designed to identify issues for improving testability. According to the information that could be acquired, the model only consists of a maturity model.

The Testability support Model has only three levels. TSM has six Key Support Areas, KSAs, analogous to the KPAs in the CMM:

- Software Engineering infrastructure.
- Project plans.
- Product information.
- Software design.
- Testware.
- Test environments.

Comparison with TMM

The description above is the only public information available and is considered insufficient to compare the TSM with the TMM usefully.

Annex F Table of characteristics of all models

	TMM (reference)	**MMAST**	**TAP**	**TCMM**
Type Model	Maturity	Estimating automation	Maturity	Maturity
Year of development	1996	1994	1995	1996
Approach	Generic	Practical	Practical	
Maturity model foundation	Paulk et al (1993b), Gelperin and Hetzel (1988), Durant (1993), Beizer (1990)	Houston (1993), Rakatin (1994), Humphrey (1989)	Paulk et al (1993b) Crosby (1979)	Paulk et al (1993b)
Number of levels	5	4	5	5
Number of KPAs	13	N.A	4	?
KPA type	One level	N.A.	?	One level
Coverage of test activities	+	--	?	?
Assessment type	Questionnaire	Self assessment	External	External
Assessment elements	Assessment Procedure, Questionnaire, Training and Team Selection criteria	Questionnaire	Questionnaire	Questionnaire
Assessment foundation	CMM, ISO, SPICE	None	?	?
Information about model that could be obtained	Articles Dissertations	Just one article	Just one article	Just one article
Ideas for MB-TMM	N.A.	Overview of costs	People KPA	None
Remarks	Dissertations difficult to obtain.	Little information available	Information about a TAP project could be obtained	Little information available

Table 3: an overview of characteristics of all models.

	TIM	**TOM**	**TPI**	**TSM**
Type Model	Maturity	Symptom	Maturity	Maturity
Year of development	1996	?	1997	?
Approach	Practical	Practical	Practical	?
Maturity model foundation	Paulk et al (1993B), Gelperin (1996a, 1996b)	None	None	?
Number of levels	5	N.A.	14	3
Number of KPAs	5	N.A	20	6
KPA type	Multiple level	N.A	Multiple level	Multiple levels
Coverage of test activities	+-	-	++	?
Assessment type	Questionnaire	External	Checklist	?
Assessment elements	Assessment guideline Questionnaire	Questionnaire	Assessment guideline Checklist Test Maturity Matrix	?
Assessment foundation	None	None	None	?
Information about model that could be obtained	Articles (a few)	Articles (a few)	Articles Book	Article (just one)
Ideas for MB-TMM	Different types of assessments	None	Improvement suggestions Checklist Test, environment Office environment Reporting Defect management Testware management	None
Remarks	Little information available			The articles[Gelperin 1993a], [Gelperin 1993b] could not be obtained.

Table 3: an overview of characteristics of all models (continued).