Development and validation of a metric based testing maturity model

Wat kunnen we in de naaste toekomst verwachten? Software zal een belangrijke rol blijven spelen in onze maatschappij. Daarmee blijft er winst te behalen door software te ontwikkelen. Of het daarbij nu gaat om maatwerk voor specifieke gebruikers of om softwarepakketten die generiek toepasbaar zijn. De kwaliteit van die software blijft doorslaggevend in het succes. SPI is gebleken een aanwinst te zijn voor het vakgebied. Als uw concurrenten er hun winstkansen mee vergroten, zult ook u moeten nagaan of er winst te behalen is.

Keynote sprekers

Jennifer Stapleton is betrokken geweest bij kwaliteitsvragen rond software en softwareontwikkeling. Zij was een van de stuwende krachten achter de organisatie van SQM (Software Quality Management), een internationale conferentie op dit gebied. Tevens is zij nauw betrokken geweest bij de ontwikkeling van DSDM. De combinatie van deze ervaringen belooft een eigenzinnige visie op het proces en de methode van softwareontwikkeling.

Martin Pol is zowel nationaal en internationaal bekend om zijn inzet om testen en de verbetering van het testproces te professionaliseren. Hij zal zeker kunnen weergeven hoe de kwaliteit van software verbeterd moet worden.

Mike Konrad en Judith Mogilensky zijn beide nauw betrokken bij de ontwikkeling van CMMI. In een onderling debat zullen zij de betekenis van CMMI voor het voetlicht brengen.

PROGRAMMA

09.30	Ontvangst
09.55	Welkomstwoord, Wilko van Asseldonk
10.15	Does Agile + Fragile ***Jennifer Stapleton***
11.00	Koffiepauze
11.30	SPIconomics *Prof. Dr. Egon Berghout*
12.15	SPI en testen ***Martin Pol***
13.00	Lunch
14.30	SPI bijdrage aan proces van waardecreatie *Drs. W. Hassoldt MBA RE, Dr. J. Spangenberg*
15.15	10 Jaar Software Process Improvement bij Philips Electronics; de aanpak en de resultaten
16.00	Theepauze
16.30	CMM : Cultuur, Mensen en Macht ? *Ir. Ino Rots*
17.15	DEBAT CMMI versus CMM ***Mike Konrad, Judah Mogilensky***
18.15	Afsluiting
18.30	Aperitief en diner
20.30	Einde programma

Aanmelden kan via de SPIder Congres site, www.euroforum.nl/e50726.html

Development and Validation of a Metrics Based Testing Maturity Model[1]

Many organizations and industries continually struggle to achieve the foundation of a sound testing process. Because testing is only marginally addressed in software process improvement models like CMM, a separate Testing Process Improvement model is clearly needed. The authors represent a consortium of companies and organizations operating in the Netherlands. The consortium was formed to develop a Testing Process Improvement model.

The framework of the model is based on the “Testing Maturity Model (TMM)”, but considerable enhancements are provided. Enhancements include additional process areas, systematic treatment of implementation actions, provision of an instrumentation repertoire including a test assessment procedure, a detailed metrics program to determine effectiveness and efficiency of the improvements and associated selection procedures for improvement actions. The resulting model is dubbed MB-TMM (Metrics Based Testing Maturity Model). Subsequently, MB-TMM is validated by means of real-life experiments, and adjusted or adapted when necessary.

This paper addresses the development approach of MB-TMM, an outline of the model, early validation experiments, current status and future outlook.

1 Introduction

Software testing is coming of age. A wide catalogue of excellent books on the subject exists, specialised journals are available, focused conferences, seminars and workshops are held, special interest groups are in place, news groups flourish, training services are offered and a certification program exists. Nevertheless, many organizations still struggle with the founding of a sound testing process. One of the reasons is that existing software development maturity models, like CMM [10], have not adequately addressed testing issues nor has the nature of a mature testing process been well defined. What is a sound testing process in the first place? How should you organise and implement test process improvement? How should you embed it into the organization? What are the consequences for the organization? In short, guidance for the process of test process improvement is badly needed, as well as a method to measure the ma-

[1] This project was subsidised by the Dutch Government (Ministery of Economic Affairs).

turity level of testing, analogous to the widely adopted Capability Maturity Model (CMM) for the software development process.

Some well known models for test process improvement are TIM (Test Improvement Model), TOM (Test Organization Maturity model), TPI (Test Process Improvement model), and TMM (Testing Maturity Model). Each of these models, of course, has its own characteristics, strengths and weaknesses.

In 1999 a rather unique initiative was undertaken: a number of industrial companies, consultancy & service agencies and an academic institute, all operating and residing in the Netherlands, decided to jointly develop, validate and disseminate a testing improvement model. This model should unite the strengths and remove the weaknesses of known improvement models.

Apart from direct and short-term benefits for the participating parties, i.e. improvement of their own testing processes and/or services, a widely accepted and supported testing improvement model has attractive longer-term advantages, like unification of testing education, facilitation of the exchange of test personnel, cost reduction, testing efficiency, etc.

This paper addresses the achievements so far. Chapter 2 describes the objectives of the model and the development approach. Chapter 3 is concerned with the conceptual basis of the model, which has lead to the framework of the model, described in chapter 4, while the structure of the model is addressed in chapter 5. Chapter 6 describes some preliminary experiments to validate aspects of the model. Finally, chapter 7 gives the current status of the model and the future outlook.

2 Objectives and Development Approach

A consortium was formed of industrial companies (Thales, formerly Hollandse Signaal Apparaten B.V., Lucent Technologies Nederland B.V. and Philips Electronics Nederlands B.V.), consultancy & service organizations (Improve Quality Services B.V., Quality House B.V.) and an academic institute (Frits Philips Institute, University of Technology–Eindhoven). The industrial partners operate in diverse and high-demanding fields including defence and civil systems, telecommunication and satellites, consumer and professional electronics. The consultancy & service partners operate in software quality, testing, and related vocational training. The academic partner is a technology research institute affiliated with the University of Technology Eindhoven, specialising in R&D for technology-intensive companies.

The development of a testing improvement model could highly benefit from this unique blend of participants. The industrial partners are the drivers for applicability, usability, concreteness, economic and business view. The consultancy & service partners emphasise the generizeability, saleability, versatility, learn-ability and commerceability aspects. The academic partner brings in the scientific foundations and counterbalances the supposed pragmatic orientation of the other partners.

The participating organizations were represented by their (senior) testing experts. In addition, two students from the University of Technology Eindhoven were available full-time to work on the model in he context of their graduation (Degree in Industrial Engineering & Management Science). In early discussions on the joint development of such a model a central question was whether start development from scratch or use an existing model as basis. And if the latter approach were taken, what would be taken as base model?

The partners required that the developed model be universally applicable (that is, not geared towards a specific type of business) and identified that the model should at least:

- Describe a coherent and generic set of steps and actions to improve a testing process
- Provide well-defined test maturity levels
- Provide an instrument to asses test maturity
- Define measurements to determine the effectiveness of improvement actions
- Recommend a metrics base to select process improvements, to track and control implementation of improvement actions and to adjust and focus the process improvements.

The model should minimally address:

- Integration of the test improvement model with software process improvements models (like CMM-SW, CMM-I)
- Institutionalisation of test methods and techniques
- Set-up of a test organization
- Methods to verify and validate system work products (e.g. review methods)
- Prioritisation and reduction methods of test sets
- Feedback mechanisms for defect prevention.

After intensive deliberations and preliminary research scans, it was agreed to begin a joint project and using an existing test improvement model, TMM (Testing Maturity Model) as basis [1][2]. The main reasons for this choice, supported by all testing experts of the consortium partners, were that this model already seems to fulfil quite some of the objectives that the consortium had in mind, that TMM reflects over forty years industry-wide software testing evolution and that TMM was designed to be a counterpart of the software process improvement model CMM [10]. However, should the joint project suggest another base model (model investigations were included in the project) the choice could be reconsidered.

Tentatively, the project was called MB-TMM (Metrics Based Testing Maturity Model), emphasising that TMM was used as starting point, and that a Metrics Base should be provided. A project proposal was compiled in which the MB-TMM development was divided in five stages:

Stage 1 is focused on extended inventory and examination of existing software improvement models, test process improvement models and an evaluation of their fit to the consortium's model objectives. In addition, best practices (in testing) and testing-related standards would be identified and collected. This stage is characterised by state-of-the art literature investigations concerning improvement models, software testing and metrics-design methodologies.

Stage 2 is the R&D stage in which the results of stage 1 are used to develop a conceptual framework for MB-TMM. Test maturity levels are defined and process areas are worked out in detail. In addition, the framework for the metrics base is established. This stage delivers the model along with coherent and generic set of steps and actions to improve a testing process.

Stage 3 is dedicated to the specification and design of the instrumentation repertoire for MB-TMM that results from work stage 2. Focal points include test assessment procedures, a detailed metrics program and procedures for selecting improvement actions.

Stage 4 is the experimental stage. MB-TMM and associated instrumentation is put into practice, with the goal to validate and evaluate (aspects of) the model and to adjust or refine it if necessary.

Stage 5 is concerned with the dissemination of the model and its associated instrumentation.

In turn, each of the partners would act as project leader for one or more of the stages (or associated sub-stages), to reflect the joint-project character of the endeavour. Assignments are made according to specific expertise and/or special interest areas. All participants review Work products.

3 Conceptual Basis of the model

The first step towards the definition of a framework for MB-TMM was an extended inventory and examination of existing improvement models and an evaluation of their fit to the consortium's model objectives [4][5]. Among the models investigated were general software improvement models like CMM and its successor CMM-I, SPICE, Bootstrap, and software testing specific models like TMM, TPI, TAP, MMAST, TCMM, TSM, TIM and TOM, including comparisons of models [9].

In addition, the literature was scanned for best test practices and test standards as a preparation for later process area definitions [6]. The literature scan also included approaches for development and application of metrics [7], as a preparation to the development of a metrics base for MB-TMM.

The comparison of TMM and other Test Process Improvement Models was of paramount interest for the MB-TMM project. Though TMM was tentatively chosen as base model, the comparison of models should justify the choice, by identifying strengths and weaknesses of models. The model comparisons as well as a real-life account of application of TMM justified the choice for TMM [8] [9].

TMM (Testing Maturity Model), developed, in 1996 at the Illinois Institute of Technology [1][2] reflects the evolutionary pattern of testing process maturity growth documented over the last several decades, as outlined in a historical model provided by Gelperin and Hetzel [3]. A definite strength of TMM is that it is founded on forty years of industrial experience with software testing. It profits from many past struggles to find a sound software testing process.

Also a very strong point of TMM is its design objective: to be a counterpart of the popular software process improvement model CMM. Software process improvement programs can use TMM to complement CMM, as CMM does not adequately address software-testing issues. On the other hand, it is also possible to improve the testing process independently, though one should be aware that maturity levels for testing and software development must remain close to each other.

TMM is a highly conceptual model. As such it fits every business environment. It leaves a lot of room for business characteristics and its testing process. This is an attractive thought but it also has the downside that TMM is not a cookbook for establishing or improving a testing process. It needs the hands and brains of an experienced test process improvement leader to implement an effective, efficient and managed test process. However, the same can be said of virtually any other improvement model.

One of the biggest weaknesses of TMM is its rather poor description. Just compare the brief journal-like style of the TMM description with the extensive SEI's improvement model. TMM's cursory description causes a number of related weaknesses. Lack of detail and insufficient explanation of terms results in inconsistent usage of those terms.

Another weakness is the relative under-representation of goals or activities for people management and the test organization. The development of a maturing test process implies the development of a maturing test organization, which has not been adequately covered by TMM.

Also missing in TMM is explicit attention for the test environment. Test environment refers to test equipment, test systems, test beds, etc. Technical software environments often require special test systems or equipment, which is quite often used by developers as well. A maturing testing process also requires a maturing management of the test environment. The test environment is paramount in testing and must therefore be addressed in any test process improvement model.

An issue overlooked or underrepresented by virtually all models, including TMM, is that at the improvement actions of higher maturity levels cannot be performed independently from other organizational entities. To improve the software process from CMM level 4 on, alignment with e.g. marketing and sales department, manufacturing department is required. To improve the software testing process, the test organization has to be aligned with the development department, marketing and sales etc. Processes get a wider and wider scope at higher maturity levels and consequently require tuning and alignment with other departments.

4 The Framework of the Model

Crucial for the further development of MB-TMM was the decision to go for a continuous or a staged model. In a continuous model every process area is addressed at every maturity level. This implies that at every maturity level all process areas are simultaneously improved. This seems to be logical: all aspects of the testing process smoothly grow in maturity at the same time. In contrast, a staged model focuses on different process areas per maturity level, although some process areas can be addressed at multiple maturity levels.

Within the MB-TMM development consortium an early preference for the continuous model approach emerged. However, during development of the model-framework it was experienced that a continuous model structure proved hard to define. A staged model was shown to be much more practical, easier to design and to implement.

The maturity levels and most of the process areas of MB-TMM are about the same as with TMM. However, there are several major differences between MB-TMM and TMM. In addition to TMM, the MB-TMM has:

- A CMM(I) like structure, including CMM(I) terminology
- Comprehensive and consistent description of process areas
- Comprehensive glossary of terms
- An extra “Test Environment” process area
- An extra “Organizational Alignment” process area
- An extension of the “Technical Training” process area with “career development”
- A section “Recommended Literature” accompanying every process area
- A metric base to measure test process improvement and to support the improvement process
- An improved assessment model.

The model with its maturity levels and process areas is given in figure 1.

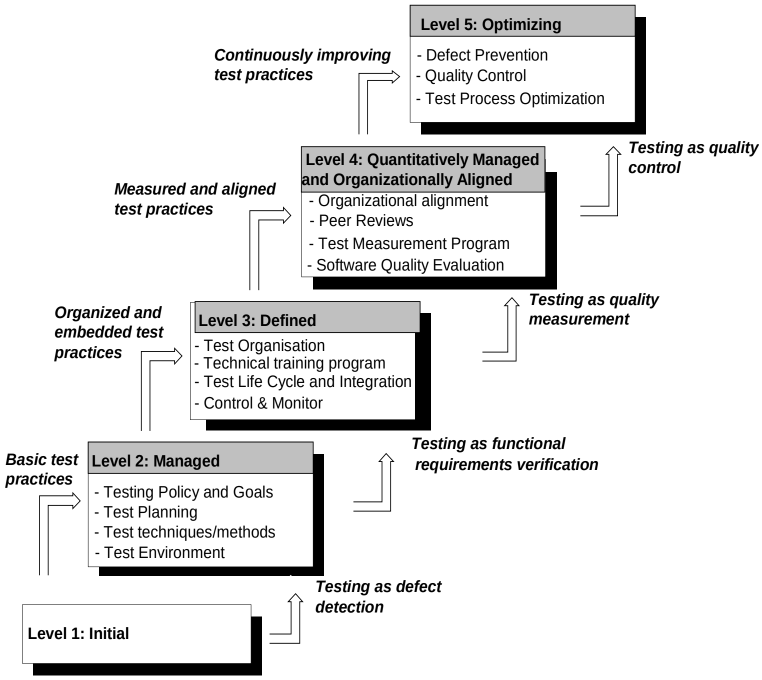

Figure 1 Maturity levels and Process Areas of the Metric Based Testing Maturity Model MB-TMM

Note that the layout of the model is very similar to CMM(I) and TMM. Just like TMM, MB-TMM is to be considered as a companion to CMM(I).

At the MB-TMM level 1, the Initial level, the main objective of testing is to show that software products work. Testing is performed in an ad hoc way, after coding is done and when time allows. Testing is a spiral of finding and correcting problems, without separation. There is a lack of resources, tools and properly trained staff. There are no process areas at this level.

By the introduction of basic test practices, a basic testing process emerges at MB-TMM level 2, the “Managed” level. The objective of testing is now defect detection. Testing and debugging are now considered different activities. Testing is still (exclusively) executing code, but is now performed in a systematic and managed way. Testing is planned, performed and documented. Tests are conducted in a dedicated test environment.

Further organization of testing and embedding into the development life cycle, allows the process maturity to rise to TMM level 3, the “Defined” level. Testing has become a real verification of functional requirements as laid down in a specification document according to a defined and repeatable process, documented in standards, procedures, tools and methods. Testing begins already at the requirements phase and continues throughout the life cycle. A test organization is in place and testing is recognised as a profession, including a career development plan and associated training program.

Measured and aligned test practices are introduced to reach TMM level 4, the “Quantitatively Managed and Organizationally Aligned “level. Testing as now considered as quality measurement of software products, in all aspects. The conditions to operate at this level are created by aligning the way-of-working with other organizational entities. Quantitative measurements and statistical techniques and methods control the testing process.

At TMM level 5, the "Optimizing" level, testing has evolved to total software product quality control, testing is a streamlined, defined, managed and repeatable process, where costs, efficiency and effectiveness can be quantitatively measured. Testing is continuously improved and fine-tuned in all aspects. Defect collection and analysis are practised with mechanical precision to prevent defects from recurring.

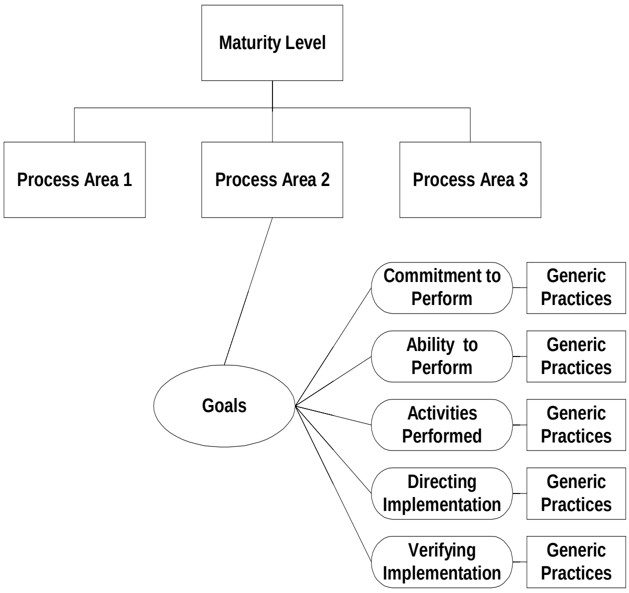

Figure 2 Structure of the Metric Based Testing Maturity Model MB-TMM

5 The Structure of the Model

MB-TMM describes process areas in a CMM-like way. The elements are described in table 1 below (including terminology used in TMM), while figure 2 shows how the elements are combined into the structure.

MB-TMM element	TMM term	Description
Maturity Level	Maturity Level	A defined stage in process improvement. Each maturity level stabilises an important part of the organization's processes
Process Areas	Maturity Goals	A group of related practices performed collectively to achieve a set of objectives, including what it does and the anticipated behavior
Goals	Maturity Sub-goals	Unique characteristics that describe what must be achieved to satisfy the purpose of a certain process area. Goals are required elements, and apply to only one process area
Common Features	Critical views	Predefined attributes that identify categories of practices and activities
Generic Practices	ATR's	Practices that apply to every process to improve the performance and control. Generic practices are identified for each common feature except for "Activities Performed"
Activities	ATR's	Expected elements that are considered important for achieving a process area. Activities only apply to the common feature "Activities Performed"
Metrics	-	Quantitative information used to track and control implementation of improvement actions, to adjust or focus process improvements.

Table 1 Description of the MB-TMM structure elements

Common Features and Generic Practices

Common Features are predefined attributes that group Generic Practices into categories. Common Features are model components that are not rated in any way, but are used only to structure the Generic Practices. Five Common Features are distinguished, structuring a total of eleven Generic Practices, as shown in table 2 below, and detailed in the remainder of this section.

Common Features	Generic Practices
Commitment to Perform	• Establish an organizational policy
Ability to Perform	• Provide resources
	• Assign responsibility
	• Train people
	• Establish a defined process
Activities Performed	• Activities
Directing Implementation	• Manage configurations
	• Measure the process results
	• Identify and involve relevant stakeholders
Verifying Implementation	• Objectively evaluate adherence
	• Review status with business management

Table 2 Overview of CommonFeatures and Generic Practices

Commitment to Perform

- Establish an Organizational Policy

The purpose of this generic practice is to define organizational expectations for the process and make these expectations visible to those in the organization that are affected. In the process area descriptions, this generic practice is abbreviated as “Policy”.

Ability to Perform

- Provide Resources

The purpose of this generic practice is to ensure that the resources necessary to perform the process as defined by the plan are available when they are needed. Resources include adequate funding, appropriate physical facilities, skilled people and appropriate tools. The interpretation of the term "adequate" depends on many factors and may change over time. Inadequate resources may be addressed by increasing resources or by removing requirements, constraints, and commitments. In the process area descriptions, this generic practice is abbreviated as "Resources”.

- Assign Responsibility

The purpose of this generic practice is to ensure that there is accountability throughout the life of the process for performing the process and achieving the specified results. The people assigned must have the appropriate authority to perform the assigned responsibilities. Responsibility can be assigned using detailed job descriptions or by living documents, such as a process plan. Dynamic assignment of responsibility is another legitimate way to perform this practice, as long as the assignment and acceptance of responsibility is assured throughout the life of the process. In the process area descriptions, this generic practice is abbreviated as “Responsibility”.

- Train People

The purpose of this generic practice is to ensure that the people have the necessary skills and expertise to perform or support the process. Appropriate training is required to the people who will be performing the work. Overview training is provided to orient people who interact with those performing the work. Training supports successful. Implementation of the process by establishing a common understanding and by imparting the skills and knowledge needed to perform according to the process. In the process area descriptions, this generic practice is abbreviated as “Training”.

- Establish a Defined Process

The purpose of this generic practice is to establish and maintain a description of the process that is tailored from the organization's set of standard processes to address the needs of a specific instantiation. With a defined process, variability in how the processes are performed across the organization is reduced; and process assets, data, and learning can be effectively shared. The descriptions of the defined processes provide the basis for planning, performing, and managing the activities, work products, and services associated with the process. In the process area descriptions, this generic practice is abbreviated as “Process”.

Activities Performed

- Activities

The purpose of this generic practice is to describe the activities that must be performed to establish the process. Typically, a set of related activities is necessary to adequately address each process area. In the process area descriptions, this generic practice is abbreviated as “Activity”.

Directing Implementation

- Manage Configurations

The purpose of this generic practice is to establish and maintain the integrity of the designated work products of the process (or their descriptions) throughout their useful life. The designated work products are specifically identified in the plan for performing the process, along with a specification of the level of configuration management. Different levels of configuration management are appropriate for different work products at different points in time. In the process area descriptions, this generic practice is abbreviated as “Configuration mgt”.

- Measure the Process

The purpose of this generic practice is to perform direct day-to-day monitoring and control of the process and to collect information derived from planning and performing the process. Appropriate visibility into the process is maintained so that corrective action can be taken when necessary. This practice is performed so that performance information can be included in the organization's process assets and made available to those who are (or who will be) planning and performing the same or similar processes. The information is stored in the organizational measurement repository and the organizational library of process-related assets. In the process area descriptions, this generic practice is abbreviated as “Measure”.

- Identify and Involve Relevant Stakeholders

The purpose of this generic practice is to establish and maintain necessary involvement of stakeholders throughout execution of the process. Involvement assures that interactions required by the process are accomplished and prevents affected groups or individuals from obstructing process execution. In the process area descriptions, this generic practice is abbreviated as “Stakeholders”.

Verifying Implementation

- Objectively Evaluate Adherence

The purpose of this generic practice is to provide credible assurance that the process is implemented as planned and satisfies the relevant policies, requirements, standards, and objectives. People not directly responsible for managing or performing the activities of the process typically evaluate adherence. As a result, credible assurance of adherence can be provided even during times when the process is under stress (e.g., when the effort is behind schedule or over budget). In the process area descriptions, this generic practice is abbreviated as "Adherence".

- Review Status with Business Management

The purpose of this generic practice is to provide business management with appropriate visibility into the process. Business management includes those levels of management in the organization above the immediate level of management responsible for the process. These reviews are for managers who provide sponsorship and overall guidance for the process, not for those who perform the direct day-to-day monitoring and control of the process. Different managers have different needs for information about the process. These reviews help ensure that informed decisions on the planning and performance of the process can be made. Therefore, these reviews are expected to be both periodic and event driven. In the process area descriptions, this generic practice is abbreviated as "Review".

6 Case Studies

Several case studies have already been carried out to both validate and elaborate identified process areas. The objective is to provide process areas with operational instruments such as methods, techniques and tools that support specific procedures of a process area. One of the instruments that has been developed is a Risk Analysis Method that supports practitioners in the process area Peer Reviews. Another example is a set of practical guidelines, that has been developed to trace requirements, and that serves practitioners in the process area control and monitoring. Both developments will be addressed briefly in the following.

The process area 'Peer Reviews'

Some partners in the MB-TMM project experience problems with inspecting (very) large documents. Inspections and reviews are expensive and often there are only limited resources available. One of the main questions is how to decide which parts of a document have to be checked? The consortium decided to develop a Risk Analysis Method that supports the identification of the critical parts of a document and the determination of the defect density of those critical parts. As such the Risk Analysis Method can be used to determine which parts of a document have to be reviewed or inspected

In fact the Risk Analysis Method is used for prioritising inspection effort. The method consists of five steps. In the first step the critical document parts are identified. Criticality is defined as the degree of impact that a severe defect in a specific part of a document can have on the final product or development process. In step 2 defect density factors are determined. Defect density factors are factors that influence the quality of a document. An example is the time pressure under that a document has been written. Examples of other factors influencing the defect density are: Amount of reuse, Use of standards and checklists, Domain experience of author, etc. In step 3 the critical pages are associated with the defect density factors. In steps 4 and 5 defect densities are calculated per page and it is determined on basis of a sampling algorithm which parts of the document have to be inspected. The Risk Analysis Method has already been applied in three experiments.

The process area 'Control and Monitor'.

The purpose of monitoring and control is to provide insight into the test project's performance so that it can be managed. The Process Area Control and monitor addresses the management of a test project by means of the controlling and monitoring of progress of the test project and the quality of the products being tested.

Control and monitoring is the process area in that among others requirement management processes are allocated. Requirement Management is the process of managing changes to the system requirements. As such it encompasses both change management and document maintenance. Concepts such as quality of requirements and configuration management of requirements have been elaborated on basis of case and literature studies. In the following we will describe these two concepts briefly.

- *Quality of requirements*

Requirements should be able to pass the Quality Gate [13]. A good requirement states something that is necessary, verifiable, attainable, and is formulated in a clear and unambiguous manner. The Quality Gate consists of a number of questions and checks that need to be performed on each requirement, in order for it to pass the quality control. NASA has developed a tool that assists in the evaluation the quality of the requirements specification document and the individual requirements [12]. The Automated Requirements Measurement (ARM) tool searches requirement documents for certain quality indicators. The tool scans the text of the specification for specific primitives and keeps track of them. Based on these findings, a judgement is made on the quality of the specification and the requirements individually. The Quality Gate (technique) and ARM (tool) were considered as useful means to perform an intake check on the requirements, to see if they are testable, and usable for the development process.

Based on the case studies and the literature research also guidelines have been developed regarding the establishment of specification and style standards. Project participants should be trained in the use of those standards.

- *Configuration management of requirements*.

The most important aspect of requirement management was considered to be the traceability. Traceability is essential for adequate change management and document maintenance.

Based on a case study three different types of traceability have been defined. The three types of traceability are respectively:

Horizontal traceability

Horizontal traceability is the possibility to trace relations between requirements of (sub)products and/or components that emerge in the development process from requirements to analysis, design and implementation (including testing).

Vertical traceability

Vertical traceability is the possibility to trace relations between (sub)products and/or components that emerge form top-down or bottom-up decompositions of requirements.

Temporal traceability

Temporal traceability is the possibility to trace relations between (sub)products and/or components and associated documentation that emerge through time.

For each of the three types of traceability guidelines have been developed to support test professionals with configuration management of test products.

7 Current Status and Future Outlook

The current status of the MB-TMM project is that the exploration phase has been finished and that the design phase is well under its way. The latter is reflected in this paper by the descriptions of the MB-TMM frame and the process areas (which are already defined up to level 4). From a content point of view a key issue in the further development of the MB-TMM will be the staged/continuous aspect. Although the MB-TMM is currently clearly a staged model of five levels, the partners in the consortium feel that they can address continuous aspects of the MB-TMM as well. To be able to do so they defined so-called 'key issues in improvement' that appear at each level. These key issues are respectively People, Organization, Technology and Process. By making clear at each level and in each process area in what way and to what extent these key issues are addressed the structure of the MB-TMM will be strengthened. Further this will serve as an aid to check the completeness of the MB-TMM, both at each level and for the whole MB-TMM. Making the key issues explicit can also be used as basis for the development of the assessment approach in the sense that testing organizations will be assessed by focussing on these key issues in particular.

Although the scope of the MB-TMM is systems testing it should be stressed here that currently most of the process area descriptions have a strong software testing orientation. This is caused by the actual interests, knowledge and skills of the consortium partners. However CMMI, which has as scope systems improvement, is still the main reference for the development of MB-TMM. In the experiments that will take place in the next months it has to become clear whether the scope of the process area descriptions has to be enlarged from software to systems and in what way this has to be done.

Current activities are the development of the metric base and the assessment approach and procedures. Regarding the metric base it has been decided that each process area will contain Goal Question Metric [11] procedures to develop a metrics program for the evaluation and determination of the effectiveness of specific testing improvement activities. With respect to the assessment approach a number of checklists have already been developed that will be validated and enriched in the experiments.

Various experiments will take place in the next months. We mention here for example the real-life experiments in that the usage of the process area descriptions will be validated in testing projects at the partners'organizations. Further, prototype instruments such as the assessment checklists will be applied in the testing departments or areas of the partners'organizations. Regarding the risk analysis method, this instrument will be validated using historical data of recently carried out review activities.

Currently the ownership of the MB-TMM is being determined. It is likely that on the short term a MB-TMM platform organization will be developed that will exploit the MB-TMM and that is responsible for dissemination (via Web and book publications), maintenance and improvement activities. This platform organization will develop close relations with relevant other organizations, institutes and/or specialists in the field of systems and software process maturity, such as SEI, IQUIP, Burnstein etc.

References

[1] Burnstein I., Suwanassart T., and Carlson C., 1996-a. Developing a Testing Maturity Model. Part 1. Crosstalk, Journal of Defense Software Engineering, 9, no. 8, 21-24. Also available on: http://www.stsc.hill.af.mil/crosstalk/1996/aug/developi.asp

[2] Burnstein I., Suwanassart T., and Carlson C., 1996-b. Developing a Testing Maturity Model. Part 2. Crosstalk, Journal of Defense Software Engineering, 9, no. 9, 19-26 Also available on: http://www.stsc.hill.af.mil/crosstalk/1996/sep/developi.asp

[3] Gelperin D. and Hetzel B., 1988, The Growth of Software Testing, Communications of the ACM, 31, no. 6, 687-695

[4] Ham, Martijn, 2001-a, MB-TMM. State of the art literature. Part 2, MB-TMM project report 12-7-1-FP-IM

[5] Ham, Martijn, 2001-b, Testing in Software Process Improvement, MB-TMM project report 12-5-1-FP-IM

[6] Ham, Martijn, Veenendaal, Erik van, 2001, MB-TMM. State of the art literature. Part 1, MB-TMM project report 12-1-1-FP-IM

[7] Ham, M. , Swinkels R., Trienekens J.J.M., 2001, Metrics and Measurement, MB-TMM project report 13-6-1-FP

[8] Jacobs J., van Moll J. & Stokes T., 2000, The Process of Test Process Improvement, XOOTIC Magazine, 8, 2

[9] Swinkels, Ron, A Comparison of TMM and other Test Process Improvement Models, MB-TMM Project Report 12-4-1-FP

[10] Paulk, M, Curtis B., Chrissis M.B. and Weber, C., 1993, Capability Maturity Model for Software, Software Engineering Institute, Carnegie Mellon University

[11] Van Solingen R., Berghout E., 1999, The Goal/Question/Metric Method, A practical guide for quality improvement of software development, McGraw-Hill

[12] Wilson W. et al, 1996, Automated Quality Analysis of Natural Language Requirements Specifications, Software Assurance Technology Center

[13] Robertson S., 1996, An Early Start to Testing; How to Test Requirements. Paper presented at EuroStar 1996

Jef Jacobs, Philips Seminconductors, Eindhoven and Jos Trienekens, Frits Philips Institute, University of Technology Eindhoven

■ Geen voordelen van architectuur zonder proces, maar andersom ook niet

Over de onbreekbare band tussen product en proces

De laatste tijd is er in diverse media veel aandacht voor ‘architectuur’ van softwareproducten. Het hebben van een goed ontwerp en dus ook een goede architectuur is noodzakelijk voor een succesvol product, maar er is meer nodig. Het hebben van een goede architectuur staat namelijk niet op zichzelf. Zonder de juiste processen in een project en/of organisatie gaat het uiteindelijk toch faliekant mis. Let op: andersom ook. Een omgeving waarin namelijk alle processen tot in de puntjes verzorgd zijn, maar welke niet beschikt over een goede product-architectuur, drijft toch op een mislukking af.

Inleiding

De relatie tussen product en proces in de software engineering lijkt een kip/ei probleem. Met een goed proces maken we echt niet altijd goede producten, maar een goed product gaat uiteindelijk ten onder als we geen goede processen hebben om het te ondersteunen. En als we al accepteren dat het een kip/ei probleem is dan rijst nog de vraag: “Wie is de kip en wie het ei?”. Waar moeten we nu beginnen? Beginnen we nu eerst met de processen, of moeten we eerst het product neerzetten? Over het eerste: het proces, is de laatste jaren veel gezegd en geschreven binnen de software proces improvement wereld (SPI). Over dat laatste: het product, wordt de laatste tijd steeds meer gesproken; met name het onderwerp ‘architectuur’ krijgt daarbij erg veel aandacht.

Softwarearchitectuur is momenteel een ‘hot’ issue. De redenen hiervoor zijn divers. Er zijn de laatste tijd een groot aantal boeken op de markt gebracht over dit onderwerp, welke de aandacht voor het onderwerp stimuleren. Daarnaast wordt steeds meer geaccepteerd dat een goed ontwerp de basis is voor een succesvol product. Bovendien worden architectuurvraagstukken gevoed vanuit de toenemende aandacht voor CBD (component based development), EAI (enterprise application integration), PBD (purchase based development) en SPL (software product lines).

Deze aandacht voor architectuur is niet geheel ten onrechte, aangezien een architectuur in toenemende mate de mogelijkheden en onmogelijkheden van een softwareproduct bepaalt. Tellen we daarbij op dat softwareproducten steeds flexibeler worden ingezet en steeds maar weer zodanig aangepast worden om aan aanvullende eisen te voldoen, dan komen we al snel tot de slotsom dat één van de succescriteria voor softwareproducten het hebben van de juiste architectuur is. Wat dit ‘juist’ nu precies inhoudt is echter de moeilijkheid. Wel duidelijk is dat dit varieert over verschillende producten, omgevingen, toepassingen, gebruikers, problemen, bedrijfsprocessen, en dergelijke.

We nemen stelling dat in de huidige praktijk, zowel in de industrie als in de wetenschap, een veel te strikte scheiding tussen product en proces wordt doorgevoerd. Beiden zijn echter onlosmakelijk aan elkaar verbonden. Een product komt tot stand tijdens een proces, en een proces is de bron van elk product. Een goed proces garandeert echter geen goede producten, maar om een goed product te maken heeft men zeker geen behoefte aan een slecht proces. Beide benaderingen om een goed product te maken, de productgerichte aanpak (middels bijvoorbeeld aandacht voor een goede architectuur) en de procesgerichte aanpak (middels het definiëren en structureren van processen), bestaan echter voornamelijk separaat van elkaar.

Vaak wordt binnen het thema softwarearchitectuur de metafoor getrokken naar de bouwwereld. Dit gebeurt onder het mom van: “Wat in de softwarebouw allemaal fout gaat, gaat in de bouwwereld wel goed.” Personen die deze laatste stelling gebruiken hebben blijkbaar nog nooit een huis laten bouwen. Wij hanteren daarom liever de metafoor van een muziekcompositie. De compositie is een stuk software, met als basis een architectuur, gemaakt door de componist en vastgelegd in de partituur. Het muziekstuk tijdens de uitvoering is het uiteindelijke systeem tijdens gebruik, met het orkest als projectteam en de dirigent als projectleider. Een mooie uitvoering komt slechts tot stand door een juiste combinatie van deze factoren. In feite is een muziekstuk in werkelijkheid ook een stuk ‘soft’-ware.

Architectuur behoeft een proces

Een goed muziekstuk, een perfecte compositie, is niet automatisch fijn om naar te luisteren. Het orkest dat het speelt, de dirigent, de concertzaal, bepalen bijvoorbeeld zeer sterk het uiteindelijke resultaat. Zo is het ook voor een softwarearchitectuur. Een architectuur kan nog zo goed zijn, toch bepalen de uiteindelijke implementatieprocessen, wijzigingsprocessen, kwaliteits-processen en dergelijke of er een goed product wordt opgeleverd. Mozart heeft bijvoorbeeld prachtige composities geschreven, maar een verzameling willekeurige muziekanten zorgt nog niet vanzelfsprekend voor een mooie uitvoering.