Evaluation of Maturity Level and Recommendations for Improvement of Software Testing Process Based on Test Maturity Model Integration (TMMi): A Case Study

Rizdiani Tri Prastiti[1*], Achmad Nizar Hidayanto[2]
[1,2]Faculty of Computer Science, Universitas Indonesia
*email: rizdiani.tri@ui.ac.id

DOI: https://doi.org/10.31603/komtika.v7i2.9628

Received: 12-07-2023, Revised: 04-08-2023, Accepted: 11-08-2023

ABSTRACT

XYZ company as one of the companies that provides Over-The-Top (OTT) services has problems related to defects that pass into the production environment caused by an ineffective testing process. This has an impact on user satisfaction as indicated by various user complaints when using the application. It is necessary to evaluate the maturity level which shows the ability to perform software testing and what recommendations can be given to improve the software testing process. Test Maturity Model Integration (TMMi) as a model for improving the software testing process has been widely known to improve the testing process and positively impact product quality. XYZ company, which is looking to improve its software testing process, uses the TMMi model as a reference to determine the maturity level of the testing process and provide best practices for the testing process. The assessment was conducted using the TMMi Assessment Method Application Requirement (TAMAR) and information was collected using the Delphi method. The assessment is carried out in the process area at the maturity level of level 2 and produces a rating value of P (Partially Achieved) so that the maturity level of the XYZ company software testing process is level 1 (Initial). Recommendations are prepared based on specific practices in the Test Planning and Test Environment process areas that still have weaknesses that must be improved to reach maturity level 2 (Managed). The process of preparing recommendations is assisted by the Deming cycle which is then validated with stakeholders whether these recommendations can be implemented according to the needs of XYZ company to improve the testing process.

Keywords: *Software Testing Process, Test Maturity Model integration (TMMi), TMMi Assessment Method Application Requirement (TAMAR), Deming Cycle*

INTRODUCTION

In the era of globalization, developed countries use intellectual property rights to drive the economy through science, technology, creativity and innovation. One of the media changes is the emergence of Over-The-Top (OTT) media, a platform built on the Internet that provides video streaming services or communication services [1]. XYZ Company provides Over-The-Top (OTT) services available on various application platforms such as Android, iOS, Website, and Android TV. To be able to compete with other OTT platform competitors, the XYZ company continues to develop its products and services on an ongoing basis in order to attract more users. In addition, the quality of products and services is continuously improved to increase user satisfaction.

The company aims to release new features every year, supported by stability and product quality that can compete with competitors. But in reality, the company is still experiencing problems related to product quality which causes a decrease in user satisfaction which is indicated by the many complaints about using the application. The existence of a defect that passes into the production environment causes users to be unable to log in and pay for subscription packages, which is a crucial feature. If this problem is not resolved, XYZ company will lose many loyal users of its application.

In the process of analyzing the problem, it was found that there were root causes in the existence of test cases that did not pass the testing process. Based on the results of interviews with the Engineering Manager of XYZ company, an ineffective testing process is one of the causes of delays in feature releases and defects that pass into the production environment. Testing tends to be carried out on an ad hoc basis without careful testing planning. There are other problems related to the test environment that cannot represent the actual environment.

According to [2] in their research on the root cause analysis of software failures, most software failures occur due to improper testing stages. Software testing is the most important stage to ensure a successful product implementation process because there is no re-checking stage after this stage. Improving the software testing process can produce more quality and effective software products [3]. To be able to find out what needs to be improved in a process, it is necessary to evaluate the process based on the application of related best practice guidelines [4]. The maturity level can indicate the condition or ability to test the software and can be seen which parts are lacking in XYZ companies in implementing software testing practices.

The guideline that has been widely used to improve software development and testing processes is to determine the maturity level of the process. Choosing the right model approach to evaluate the test process is a critical success factor and is not easy for process improvement testing [5]–[7]. Previous studies have shown that applying the Test Maturity Model integration (TMMi) can improve the testing process and positively impact product quality, test engineering productivity, and development cycle time effort [8]. This is also supported by other previous studies that have used the TMMi model to improve the software testing process [9]–[11]. This paper aims to provide an understanding of how the results of evaluating the maturity level of the software testing process with the TMMi model and what recommendations can be given to XYZ company to improve the software testing process.

The paper is structured into four sections. Section 2 describes a research methodology. Section 3 gives an explanation of the result and discussion including the assessment process, recommendation arrangement, and validation results with relevant stakeholders in the XYZ company along with how the recommendations are implemented in practice. Finally, in Section 4 we conclude our paper.

METHOD

1. Research Design

The approach used in this study is a qualitative approach with an applied research method. Research with a qualitative approach method is one type of method to describe, explore and understand the specific meanings used to research natural objects [12]. In the data collection and analysis stage, information was collected using the Delphi method through questionnaires, interviews, and observation of documents or artifacts that support the assessment process [13]. Four stakeholders for the software testing process, namely the Head of Engineering, Engineering Manager, and two Quality Assurances, were asked to fill out a questionnaire containing questions related to the practices carried out in the testing process at XYZ company.

The results of the questionnaire were then concluded by the researcher and sent back to the interviewees to consider whether the conclusions were in accordance with the answers to the questionnaire that had been filled out beforehand.

Based on the scope of the research, this research is classified as a type of case study research that focuses on research on a particular case within the scope of the organization or individual [14]. At the recommendation preparation stage, there is a validation process through interviews with Head of Engineering of XYZ company to ensure that the recommendations can be applied to the company. Preparation of recommendations for improvement is sorted based on priority of interest and test flow and adjusted for implementation based on the development process that applies to the company.

2. Research Object

Product development in XYZ company is carried out using an agile scrum methodology. Product development projects are divided into three project groups, namely: feature development, partnerships, and stability and scalability.

The testing process is carried out by the developer and Quality Assurance (QA). Unit tests are also performed by colleagues in a team of developers on the same platform on a regular basis. If the risk of the deployment to be carried out is high enough, the backend team developer will carry out a load test to test the maximum load that can be accepted in the production environment. QA performs Black Box Testing manually based on the Business Requirement Document (BRD) prepared by the Product Manager, designs created by the UI/UX or Product Designer, and technical documents prepared by the Backend Developer. System Integration Testing (SIT) and User Acceptance Testing (UAT) are also carried out if the project to be released is a large and critical product. Regression testing and sanity testing are carried out before and after deployment to ensure the latest features are running properly and do not interfere with existing features.

3. Research Instrument

Evaluation of the software testing process is carried out with 3 sample projects which are representative of 3 project groups and all platforms developed by XYZ company. The assessment starts from TMMi Level 2 (Managed) to measure how far the capability of the software testing process is in a project or company. The following in Table 1 is an assessment component based on the process area at TMMi level 2.

Table 1. Assessment Component

TMMi Component	Code	Process Area
Process Area TMMi Level 2 [15]	PA 2.1	Test Policy and Strategy
	PA 2.2	Test Planning
	PA 2.3	Test Monitoring and Control
	PA 2.4	Test Design and Execution
	PA 2.5	Test Environment

At TMMi level 2, there are hundreds of sub-practices that are the basis for assessing whether TMMi practices are carried out which are contained in specific practices. The dozens of specific practices form dozens of specific goals which are divided into 5 process areas. A specific goal describes the unique characteristic that must be present in a sample of test projects to satisfy the process area. Then in each process area, there is 1 generic goal consisting of 10 generic practices. A generic goal describes the characteristics that must be present to institutionalize the processes that implement a process area [15]. The following is the number of TMMi level 2 components which can be seen in Figure 1.

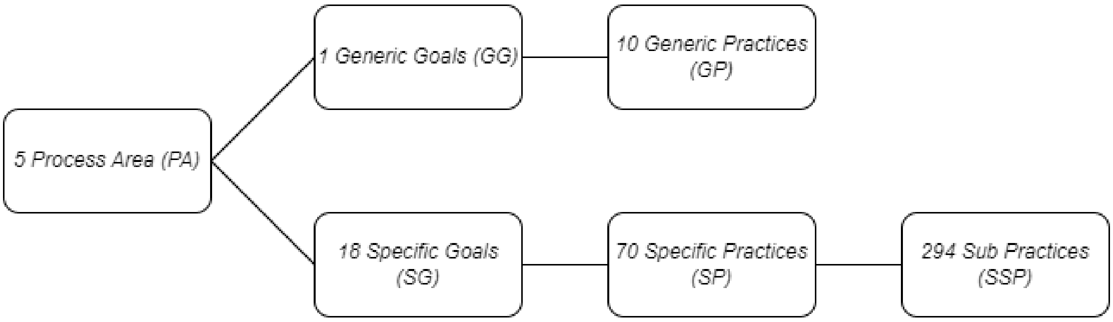

Figure 1. Number of Level 2 TMMi Components

a. Assessment Process

The stages of the assessment process can be seen sequentially in Figure 2.

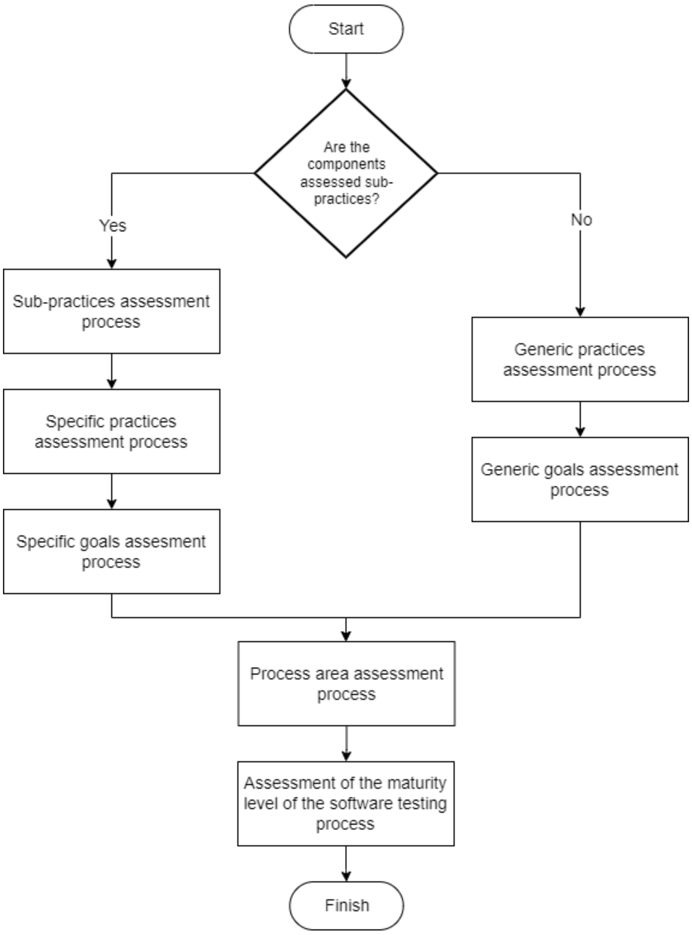
Start
Are the components assessed sub-practices?
Yes
No
Sub-practices assessment process
Specific practices assessment process
Specific goals assesment process
Generic practices assessment process
Generic goals assessment process
Process area assessment process
Assessment of the maturity level of the software testing process
Finish

Figure 2. Assessment Process Stage

The assessment is carried out based on the sub-practices and generic practices in each process area which then becomes a questionnaire to ensure that the practices are applied to the software testing process or are not included in the scope of the testing process assessment. The results of the sub-practice and generic practices assessment will indicate whether the practices have been carried out, implemented but not consistent, not implemented, or the practice is not included in the scope of the assessment of the testing process. The assessment is measured by the status of "Yes", "Partial", "No", and “Not Applicable”. Then the results of the assessment of the sub-practices and generic practices will be accumulated and the average calculated which will become the basis for evaluating other components which results in a rating consisting of several levels in accordance with TAMAR provisions. If the results of the assessment show a maturity level above level 2, then the process is continued until the maturity level is found.

RESULTS AND DISCUSSION

1. XYZ Company Maturity Model

The assessment begins with an assessment of sub-practices for each specific practice (SP) and generic practices (GP) in the five process areas. Figure 3 shows the results of the assessment of sub-practices and generic practices in the form of a diagram showing whether the TMMi practices have been implemented thoroughly and consistently (Yes), applied but inconsistently (Partial), practices have not been implemented at all (No), and are not applicable to project samples as well as the entire test.

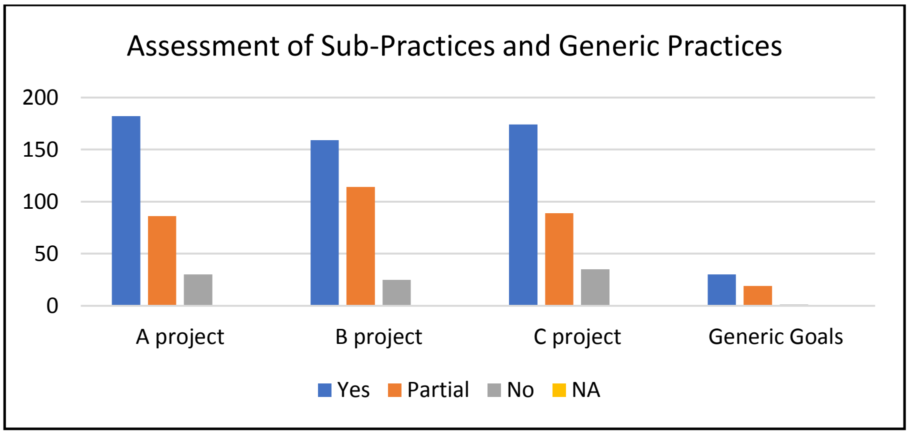

Figure 3. Assessment of Sub-Practices and Generic Practices in 5 process areas of TMMi Level 2

In Figure 3, it can be seen that TMMi's practices have been widely applied in the company's testing process, but there are still many practices that have not been consistently implemented and have not been implemented at all. After knowing the value of sub-practices and generic practices, an assessment of specific goals and generic goals is carried out which is calculated based on the average value of all practices. Table 2 is a summary of the overall assessment results for each specific and generic goal in the five process areas contained in TMMi maturity level 2.

Table 2. Assessment Result of Process Area TMMi Level 2

TMMi Components		**Rating**
PA 2.1	**Test Policy and Strategy**	**L**
SG 1	Establish a Test Policy	L
SG 2	Establish a Test Strategy	L
SG 3	Establish Test Performance Indicators	F
GG 2	Institutionalize a Managed Process	L
PA 2.2	**Test Planning**	**P**
SG 1	Perform a Product Risk Assessment	L
SG 2	Establish a Test Approach	F
SG 3	Establish Test Estimates	F
SG 4	Develop a Test Plan	L
SG 5	Obtain Commitment to the Test Plan	P
GG 2	Institutionalize a Managed Process	L
PA 2.3	**Test Monitoring and Control**	**L**
SG 1	Monitor Test Progress against Plan	L
SG 2	Monitor Product Quality against Plan and Expectations	L
SG 3	Manage Corrective Actions to Closure	F
GG 2	Institutionalize a Managed Process	F
PA 2.4	**Test Design and Execution**	**L**
SG 1	Perform Test Analysis and Design using Test Design Techniques	L
SG 2	Perform Test Implementation	L
SG 3	Perform Test Execution	L
SG 4	Manage Test Incidents to Closure	F
GG 2	Institutionalize a Managed Process	L
PA 2.5	**Test Environment**	**P**
SG 1	Develop Test Environment Requirements	L
SG 2	Perform Test Environment Implementation	P
SG 3	Manage and Control Test Environments	L
GG 2	Institutionalize a Managed Process	L

.

There are several weaknesses in PA 2.2 (Test Planning) and PA 2.5 (Test Environment) which cause the process area to get a P (Partially Achieved) score such as the specific goals "Obtain Commitment to the Test Plan" and "Perform Test Environment Implementation" which get the lowest value. Several testing practices such as documenting testing procedures based on standard testing procedure specifications and reviewing procedure specifications along with test plans with stakeholders have not been carried out consistently at XYZ companies. Test planning has not been fully carried out which causes the testing process to be less effective. The test environment conditions are not exactly the same as the actual environment so the QA or the person doing the test cannot replicate defects in the production environment in the test environment.

The process areas PA 2.1 (Test Policy and Strategy), PA 2.3 (Test Monitoring and Control), and PA 2.4 (Test Design and Execution) score L (Largely Achieved) indicate that the practices and processes are systematic and widespread. However, there are still some minor weaknesses in the distribution, application, or results of practices and processes.

Furthermore, the maturity level assessment process is carried out by taking the lowest rating value from all process areas at that maturity level as shown in Table 3.

Table 3. Maturity Level of Software Testing Process at XYZ Company

TMMi Components			**Rating**
Maturity Level	**2**	**Managed**	**P (Partially Achieved)**
Process Area	PA 2.1	Test Policy and Strategy	L (Largely Achieved)
	PA 2.2	Test Planning	P (Partially Achieved)
	PA 2.3	Test Monitoring and Control	L (Largely Achieved)
	PA 2.4	Test Design and Execution	L (Largely Achieved)
	PA 2.5	Test Environment	P (Partially Achieved)

In Table 3, it can be seen that the lowest rating value is P (Partially Achieved). Therefore, the maturity level of the software testing process at XYZ company for maturity level 2 gets a P rating (Partially Achieved). The P (Partially Achieved) value indicates that in maturity level 2, the practices and processes are incomplete, not widespread, or inconsistent in their application or results [16]. That way, the maturity level of XYZ company's software testing process is one level below the maturity level that is being assessed, which is level 1 (Initial).

2. Software Testing Improvement Recommendations

After knowing the maturity level of the testing process achieved by the XYZ company, it is necessary to prioritize the testing process improvement to determine the stages of improvement that must be carried out to improve the software testing process. Improvements are needed especially in the PA 2.2 (Test Planning) and PA 2.5 (Test Environment) process areas in order to reach maturity level 2 (Managed). Table 4 shows one example of the recommendations for improving the testing process at XYZ company in specific practices 5.1 (Review test plan) in process area 2.2 (Test Planning) by using the Deming cycle.

Table 4. Recommendations for Improvements Based on Deming Cycle

Test Planning	**SP 5.1 Review Test Plan**
Plan	
Review the test plan (and possibly other plans that affect testing) to achieve and understand test commitments. Work product: Test plan review log	
Do	
Organize reviews with stakeholders to facilitate their understanding of the test commitments.	
Check	
Manage monitoring and control of the test plan review log document to ensure that the process of reviewing the test plan and stakeholder commitment in the testing process has been carried out regularly.	
Act	
Reschedule test plan reviews with stakeholders to facilitate their understanding of test commitments.	

3. Recommendation Validation and Priority

The recommendations that have been prepared are then validated by the Head of Engineering the highest stakeholder in the field of IT operations to find out whether the recommendations given can be applied to the company and whether the implementation of the recommendations can be useful for improving the software testing process. Based on the validation results, recommendations are applicable and relevant to the software testing process carried out at XYZ company.

The recommendations given are readjusted to the testing process that has been going on and the availability of resources that support the practice of these recommendations.

Recommendations are applied in order of priority based on an analysis of the risk of improvement and the resulting impact of each draft of the test process improvement recommendations. The following is an action plan for implementing recommendations based on priority order according to the series of testing processes listed in Table 5.

Table 5. Priority Based Implementation Action Plan

Priority	Specific Practices	Action Plan
1	SP 5.2 Reconcile work and resource levels	Adjustments are made to the available resources along with the test estimates when planning the test
2	SP 5.3 Obtain test plan commitments	Assign stakeholders to each test task based on task responsibilities in the management tool and Work Breakdown Structure (WBS).
3	SP 2.3 Specify test environment intake test procedure	Develop intake test procedures such as the requirements of the test environment and test cases to be carried out to test the test environment, together with the preparation of feature test cases to be tested in the planning process.
4	SP 5.1 Review Test Plan	Conduct a review of the test plan along with the progress of the test on the WBS discussion and checkpoint discussion which results in documentation of the test plan.
5	SP 2.1 Implement the test environment	Adjust the test environment based on the needs and acceptance criteria before conducting the test.
6	SP 2.4 Perform test environment intake test	Perform an intake test on the test environment before performing feature testing.

CONCLUSIONS

Based on the results of the TMMi level 2 process area assessment, the value of the maturity level of the software testing process at XYZ company for maturity level 2 gets a P (Partially Achieved) rating indicating that the practices and processes are incomplete, not widespread, or inconsistent in application or result. Therefore, the maturity level of the company's software testing process is one level below the maturity level assessed, namely level 1 (Initial).

Improvements are needed in the software testing process, especially in the PA 2.2 (Test Planning) and PA 2.5 (Test Environment) process areas in order to reach maturity level 2 (Managed). The recommendations that have been prepared are expected to solve problems related to ineffective testing processes. Evaluation of the implementation results is necessary and if the recommendations that have been implemented can solve problems and improve the testing process to be more effective, these recommendations can be used as testing standards and as a basis for increasing the next level of maturity.

The research process can be used as a reference for research on the maturity level of testing on software testing units or organizations in similar media industries or other industries.

ACKNOWLEDGEMENT

Thank you to the XYZ Company which has helped a lot in trying to obtain the data needed.

REFERENCES

[1] C.-J. Liu and Y.-F. Chuang, “From Sluggish to Brisk: An analysis of Taiwan′s cable TV digitalization policy,” Telecomm Policy, vol. 39, no. 11, pp. 980–995, Dec. 2015, doi: 10.1016/j.telpol.2015.09.003.

[2] S. Dalal and R. S. Chhillar, “Empirical study of root cause analysis of software failure,” ACM SIGSOFT Software Engineering Notes, vol. 38, no. 4, pp. 1–7, Jul. 2013, doi: 10.1145/2492248.2492263.

[3] J. S. Collofello, Zhen Yang, J. D. Tvedt, D. Merrill, and I. Rus, “Modeling software testing processes,” in Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications, Scottsdale: IEEE, Mar. 1996, pp. 289–293. doi: 10.1109/PCCC.1996.493647.

[4] E. van Veenendaal and J. J. Cannegieter, Test Maturity Model integration (TMMi) Results of the first TMMi benchmark – where are we today? 2012.

[5] M. A. T. Laksono, E. K. Budiardjo, and A. Ferdinansyah, “Assessment of Test Maturity Model: A Comparative Study for Process Improvement,” in Proceedings of the 2nd International Conference on Software Engineering and Information Management, in ICSIM 2019. New York, NY, USA: Association for Computing Machinery, 2019, pp. 110–118. doi: 10.1145/3305160.3305203.

[6] K. Rungi and R. Matulevičius, “Empirical Analysis of the Test Maturity Model Integration (TMMi),” in Information and Software Technologies, T. Skersys, R. Butleris, and R. Butkiene, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 376–391.

[7] W. Afzal et al., “Software test process improvement approaches: A systematic literature review and an industrial case study,” J Syst Softw, vol. 111, pp. 1–33, 2016.

[8] E. van Veenendaal, V. Garousi, and M. Felderer, “Motivations for and Benefits of Adopting the Test Maturity Model integration (TMMi),” in 14th International Conference on Software Quality, Vienna, 2022, pp. 13–19. doi: 10.1007/978-3-031-04115-0_2.

[9] Md. H. Rahman, Z. Rahman, Md. Al - Mustanjid, M. S. Uddin, and M. H. R. Jany, “Software Process Improvement Based on Defect Prevention Using Capability and Testing Model Integration in Extreme Programming,” in ICONCS 2020: Cyber Security and Computer Science, Ithaca: Cornell University Library, arXiv.org, Jul. 2021, pp. 270–279. doi: 10.1007/978-3-030-52856-0_21.

[10] A. Unudulmaz and O. Kalıpsız, “TMMI Integration with Agile and Test Process,” in Proceedings of the Evaluation and Assessment in Software Engineering, New York, NY, USA: ACM, Apr. 2020, pp. 375–378. doi: 10.1145/3383219.3386124.

[11] V. Garousi and E. van Veenendaal, “Test Maturity Model Integration: Trends of Worldwide Test Maturity and Certifications,” IEEE Softw, vol. 39, no. 2, pp. 71–79, Mar. 2022, doi: 10.1109/MS.2021.3061930.

[12] J. W. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, 4th ed. SAGE Publications, Inc, 2014.

[13] D. Barrett and R. Heale, "What are Delphi studies?," Evidence Based Nursing, vol. 23, no. 3, pp. 68–69, Jul. 2020, doi: 10.1136/ebnurs-2020-103303.

[14] Z. A. Hasibuan, Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi. Konsep, Teknik, Dan Aplikasi. Universitas Indonesia, 2007.

[15] E. van Veenendaal and B. Wells, Test Maturity Model integration TMMi: Guidelines for Test Process Improvement. UTN Publishers, 2012.

[16] TMMi Foundation, TMMi Assessment Method Application Requirements (TAMAR) Version 2.0. TMMi Foundation, 2009.