European Scholar Journal (ESJ)
Available Online at: https://www.scholarzest.com
Vol. 1 No. 4, December 2020,
ISSN: 2660-5562

QUALITY MANAGEMENT PRACTICES IN SOFTWARE INDUSTRY: AN OVERVIEW OF FRAMEWORK FOLLOWED FOR SOFTWARE QUALITY MANAGEMENT

Dr.Karuna Devi Mishra,
Dr. Sudarsanam S.K

Article history:

Received November 28th 2020
Accepted: December 7th 2020
Published: December 22th 2020

Abstract:

The quality management process brings out an art of managing all activities for getting a desired level of product performance leading to product excellence .Quality is a desirable attribute to all products. In 4.0 industrial era all activities will be imbibed with the software and controls making software quality more evident A look into software quality management process was promoted right from gen 3.0 which saw the introduction of computers both hardware and software for performing all activities . Thus desirably bring out the concept of the activities of management into the software industry focusing on the processes which are definitely varied and not the same as the industrial processes hence an in-depth study in the software quality process become the integral part of the research in the areas which focus on the quality of the software which is mostly based on the testing carried out during software development process and its life cycle,

Today the biggest challenge faced by the Small and Medium Enterprises (SME's) are how to adapt, use and implement the appropriate software and right metrics at affordable cost. Therefore to achieve benefit of software testing under limited resources, it becomes necessary to identify the best software testing practices and create a mapping between various existing software methods and tools. This can be achieved by analyzing current testing practices and identifying the improvement potential.

The focus of this paper is to understand the Quality Management process specifically followed in the software industry. 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 comprehensive test process frame work. Several models are studied on goals, structure, key process areas and its assessment procedure.

Keywords: Quality Management, software quality management process, Process Improvement models,

INTRODUCTION

The quality management process brings out an art of managing all activities for getting a desired level of product performance leading to product excellence .Quality is a desirable attribute to all products. In 4/0 industrial era all activities will be imbibed with the software and controls making software quality more evident A look into software quality management process was promoted right from gen 3.0 which saw the introduction of computers both hardware and software for performing all activities . Thus desirably bring out the concept of the activities of management into the software industry focusing on the processes which are definitely varied and not the same as the industrial processes hence an in-depth study in the software quality process become the integral part of the research in the areas which focus on the quality of the software which is mostly based on the testing carried out during software development process and its life cycle, Economics of "software testing" is to determine and predict the defects early by using predictable models and implying strategies and different test methodologies to identify those at early stages. Psychology of software testing is to destructively test the application by identifying as many exceptional or out of the box scenarios or sometimes called as the third vision. Software Testing is a systematic activity but it also involves economics and human psychology. – Glenford J. Myers

Today the biggest challenge faced by the Small and Medium Enterprises (SME's) are how to adapt, use and implement the appropriate software and right metrics at affordable cost. Therefore to achieve benefit of software testing under limited resources, it becomes necessary to identify the best software testing practices and create a mapping between various existing software methods and tools. This can be achieved by analyzing current testing practices and identifying the improvement potential.

"According to an OECD Report 2014 titled 'Small and Medium-sized Enterprises: Local Strength, Global Reach' Small and Medium Enterprises (SMEs) account for over 95% of firms and in reality the Small and medium software companies comprise the majority of the software industry worldwide. In order for these small software companies to survive in a highly competitive marketplace, they must produce high quality software that will ensure a sustainable business model. As new technologies and globalization playing a vital role, the potential contribution of smaller firms is enhanced. [1]

Discussing about the significance of Testing Galin [2] explains in his book 'Software Quality Assurance "- From theory to implementation", he explicitly explains that

"Software testing is a formal process carried out by a specialized testing team in which a software unit, several integrated software units or an entire software package are examined by running the programs on a computer. All the associated tests are performed according to approved test procedures on approved test case." [3]
He further states that "testing plays a central role in quality assurance activities of many organizations and finding one metric which is that is appropriate, useful and cost effective has been the greatest key challenge for SME's across the industry." It is observed that an efficient testing practice is vital to the quality of the developed product and to reduce the overall development expenses and emphasis that software quality has a direct relationship with software testing; hence testing is an important phase of the software development life cycle. [4]

Whereas Glenford J. Myers [5] says "Software Testing is a systematic activity but it also involves economics and human psychology. Economics of software testing is to determine and predict the faults of the system early by using foreseeable models and applying structured test strategies and test methodologies to discover those at early phases of the software development life cycle. Psychology of testing is to destructively test the application by identifying as many exceptional or out of the box scenarios or sometimes called as the third vision". [6]

A set of good test scenarios evaluates every possible permutations and combinations of a program during ideal conditions. In addition, Software Test Engineer needs the proper vision to successfully test a piece/whole application to comply with the Standards and the Quality.

According to Perry [7] brings to the fore the cost and budget factor while and opines that about 24% of the overall software budget and 32% of project management budget is allocated for testing."
Software testing is the cost centre activity in the software development life cycle. Furthermore, inadequate software testing usually leads to major risks and consequences. For example, a 2002 report Tassey (2002) [8] by the American National Institute of Standards and Technology

(NIST) reported that the negative economic impacts of lack of software testing infrastructure in the United States alone amounts to $62 billion USD per year. Needless to say, there are similar challenges in other countries.
The focus of this thesis is to integrate the best processes of each test model and implement into a single frame work for Hassel free implementation of Quality management processes. 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 comprehensive test process frame work. Several models are studied on goals, structure, key process areas and its assessment procedure.

REVIEW OF LITERATURE:

This literature of review gives an in-depth understanding of the previous research and findings. During the course of this research it has been established that there are a number of different metrics for software test planning and test design processes. There are many attributes in multiple categories for software test planning and test design processes for each of these attributes, different existing measurements that has been studied during the course of research suggests that there is a need for consolidation of these measurements. Therefore in this thesis the researcher will present the consolidation of measurements that is intended to provide an opportunity of Small and Medium Enterprises (SME's) to consider adoption of a single process to software testing. This thesis aims to bring out current software testing metrics and their benefits after a comparative analysis.

The survey to study the software testing practices in Australia by Reed. K. [10] provided good insights of software testing practices useful to design this research study.

Another recently published research study by Sundmark et al (2010) [11] presents results of an industrial survey on contemporary aspects of software testing using qualitative and quantitative methods.

Their study gives crucial information about discrepancies observed between the current practices and the perceptions of respondents which could prove beneficial in shaping future research on software testing, however the explanations for these observed discrepancies were provided based on researchers assumptions, or in some cases the explanations were not clear. In this case study the observed patterns in perception of respondents will be presented and an explanation for the observed anomalies or discrepancies will be explained by using qualitative data.

RESEARCH QUESTIONS

This part of the paper aims at proposing an integrated framework for software testing that means compil By addressing the following research questions

1. What are the different methods and practices being employed for software testing?
2. What software testing activities are performed in each of these processes?
3. What are benefits of each frame work?

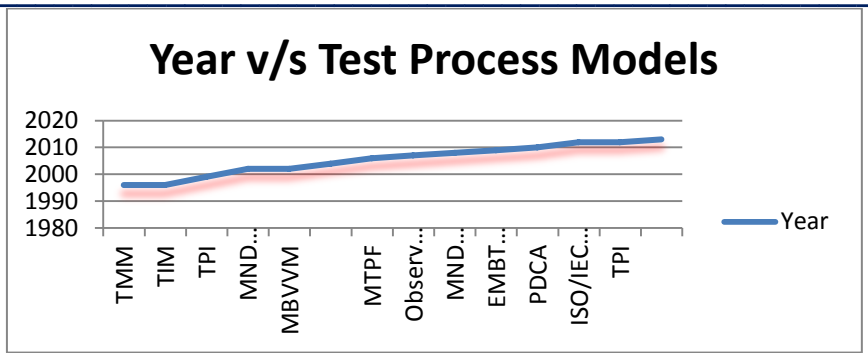
Frame work followed in **Software testing Process:**

It is imperative to have a broader perspective of software testing since there have been appalling effects of software failures. As a researcher it is paramount too analyze the backdrop to failures that has jeopardized many projects and organizations. This concept is highlighted by Peter Neumann's (1990) through column in which he has alerts the science and software fraternity on the imperative failures through his column 'Risks to the Public' in ACM's Software Engineering Notes magazine. He postulates many occurrences due to software failures varying in their intensity and impact. [i]

As explained by Tassey, G. (2002), he explains the economic impacts of improper software testing and emphases the fact that software testing is essential for quality assurance in order to establish confidence on the product and successful execution of software in the future, National Institute of Standards & Technology, technical report [ii].

The field of software engineering possesses a number of dimensions. On one axis is the development methodology. Here we refer to methodology as the software development life cycle followed, whether it is based on traditional waterfall or an iterative approach. The second axis refers to software engineering technologies which have evolved in the form of assorted programming paradigms and software architectures. We write our programs using structured programming, object-oriented or aspect-oriented programming approaches or others and design our software systems using distributed, component based or service-oriented architectures etc. On the third side we have the kind of application system to which our software will be serving. Examples are information systems, embedded systems, or communication systems etc. Figure 1.0 visualizes these dimensions. It is summary of all software testing; the chronology gives insights to the software testing methodology. Each of these SE dimensions involves peculiarities which pose special requirements on software testing. Although a meta-level generic testing process may fit any of these contexts, these three dimensions will warrant some corresponding testing considerations at lower levels of test process abstractions.

Chronology of Test Process		
1996	TMM	Testing Maturity Model (TMM) introduced in 1996 as the first model of its kind. It was followed by Test Process Improvement (TPI) model peering in 1997
1999	TPI	TPI model is based in practice and follows a structured test methodology. TPI is considered to be an objective one.
2009	EMBT TPI	The model has been designed in the context of structured high level testing. It is strongly linked with the Test Management Approach (TMap) [Pol et al., 2002] test methodology. The model elements include several key areas, each with different levels of maturity.
2010	PDCA	PDCA (plan-do-check-act, sometimes seen as plan-do-check-adjust) is a **repetitive** four-stage model for continuous improvement (CI) in business process **management**. The PDCA model is also known as the Deming circle/cycle/wheel, **Shewhart cycle**, control circle/cycle, or plan–do–study–act (PDSA).
2012	TPI	The 20 key areas within TPI are organized by means of the four cornerstones of structured testing as defined by TMap: life cycle, organization, infrastructure and techniques. Level of achievement relevant to these key areas is defined through maturity levels
2013	PDCA evidence based	PDCA (plan-do-check-act, sometimes seen as plan-do-check-adjust) is a repetitive four-stage model for continuous improvement (CI) in business process management. The PDCA model is also known as the Deming circle/cycle/wheel, Shewhart cycle, control circle/cycle, or plan–do–study–act (PDSA).
2013	TMMI	TMMi Foundation, marking the beginning of an industry-wide roadmap for implementing software quality management into the application development lifecycle. TMMi is a non-commercial, organization independent test maturity model. With TMMi, organizations can have their test processes objectively evaluated by certified assessors, improve their test processes and even have their test process and test organization formally accredited if it complies with the requirements.



Testing Maturity Model (TMM) introduced in 1996 as the first model of its kind. It was followed by Test Process Improvement (TPI) model peering in 1997. In the same year, another related approach Test improvement Model (TIM) Ericson et al., (1997) [iii] was published in an article which later disappeared for further making no serious impact and significant appreciation. After about two years later approaches similarly introduced in short articles, one in 2002 named as Metrics-based Verification & Validation Maturity Model (MB - V 2M2) Jacobs and Trienekens, (2002) [iv].

However, on similar lines another model was proposed Chernak (2004) Test Process Assessment Model (TPAM) met the same fate as TIM. On these similar maturity model specifically for the inspection process, called Inspection Capability Maturity Model (ICMM) [v]Kollanus (2005), was developed in 2005. The latest well organized and detailed development summarizes time-line of these test process evaluation and improvement models. [vi] TIM, TPAM, and MB -.V 2M2 appear to have vanished from literature probably due to their insignificance or incompleteness. This research work provides a full framework that can be used as a reference model for carrying out the test process improvement. TMMi (1999) [vii] provides an excellent reference model to be used for test assessments. TPI was the first model of its kind. It was followed by Test Process Improvement (TPI) model Pearing in 1997.

In the same year, another related approach The TPI model is based in practice and follows a structured test methodology. TPI is considered to be an objective one. By means of checkpoints it is possible to determine the levels of key areas that a test process is on. The different maturity levels and key areas and their dependencies are presented in the Test Maturity Matrix. Also, the improvement suggestions can be used for improvement actions. However, attention should be paid on the fact that the use of the TPI model does not automatically lead to good analysis of the current and required situation and to improved test process. The model should be seen as a tool for structuring the improvement of the test process and also for better communication in the organization. Regardless of the model used, improvement of the test process demands a high degree of knowledge and expertise of the people involved.

In addition, the TPI model can support determining improvement activities. The TPI model contains the following parts:

- Maturity model
- Test maturity matrix
- Checklist

TPI Next

In TPI® Next, process management at the organization level is through enablers, where process areas focus on continuous process improvement and knowledge management This model follows test maturity matrix to assess the maturity of a testing project / organization • TPI Next links each key area to the relevant aspects of the SDLC process through the concept of 'Enabler' and recognizes that testing must be executed early in the SDLC

Test Process Improvement (TPI) Koomen and Pol (1999) [viii]model is an industrial initiative to provide test process improvement guidelines based on the knowledge and experiences of a large number of professional testers. The first release of this model appeared in 1997. The model has been designed in the context of structured high level testing. It is strongly linked with the Test Management Approach (TMap) [Pol et al., 2002] test methodology. The model elements include several key areas, each with different levels of maturity.

Today the biggest challenge faced by the Small and Medium Enterprises (SME's) are how to adapt, use and implement the appropriate software and right metrics at affordable cost. Therefore to achieve benefit of software testing under limited resources, it becomes necessary to identify the best software testing practices and create a mapping between various existing software methods and tools. This can be achieved by analyzing current testing practices and identifying the improvement potential.

TPI Next and TMMi have strengths in different areas. TPI Next scores in supporting business drivers, grouping practices across process areas, scalability, having adequate implementation guidance and a standardized assessment method. Based on the comparison of rating against identified parameters we observe that seen that TPI Next is better suited in terms of ability to support majority of the parameters critical for organizations. TPI Next can be tailored to take into account the business drivers of an organization, commonly indicated as a combination of result, risk, cost and time

TEST MANAGEMENT APPROACH- TMAP

The Test Management Approach (TMap) has been developed by a Dutch firm Sogeti. A detailed description of the approach can be found in [Pol et al., 2002]. The TMap approach primarily focuses on structured testing and provides answers to the what, when, how, where, and who questions of software testing, Van Veenendaal and Pol, (1997). [ix]It is founded on four cornerstones; L a development process related life cycle model for the testing activities O solid organizational embedding I the right resources and infrastructure T usable techniques for the various testing activities Relating to each of these four aspects, TMap provides guidelines on objectives, tasks, responsibilities, deliverables and related issues. For example, the life cycle model (L) contains a sequence of testing activities which operate in parallel to the software development life cycle phases

Sogeti (2006) provides genuine assurance across testing projects TMap® next was launched with over 30 years' experience and methodology development. This means that by implementing a defined structured framework to the end-to-end test process, defects are identified earlier, timelines reduced by at least 30% and overall costs driven down. [x]

Test Process Assessment Model, TMAP can be used in conjunction with the Capability Maturity Model® (CMM®) Level 2 and Level 3. TPAM is fully consistent with the CMM structure. It presents the test process using three key process areas and defines their process goals and practices.

What is TMap® NEXT?

TMap® NEXT is an approach to structured testing. TMap NEXT was published in 2006 and it is still the standard way of testing for many process oriented organizations. It has the following advantages:

- it delivers insight into, and advice on, any risks in respect of the quality of the tested system
- it finds defects at an early stage
- it prevents defects
- the testing is on the critical path of the total development as briefly as possible, so that the total lead time of the development is shortened
- the test products (e.g. test cases) are reusable
- The test process is comprehensible and manageable.

The 4 essentials

The specific TMap® content of a structured test approach can be summarized in four essentials.

1. TMap® is based on a business-driven test management (BDTM) approach
2. TMap® describes a structured test process
3. TMap® contains a complete tool box

ISO/IEC/IEEE 29119

The purpose of the ISO/IEC/IEEE 29119 Software Testing standards is to define an internationally-agreed set of standards for software testing that can be used by any organization when performing any form of software testing. ISO/IEC/IEEE 29119-5 defines Keyword-Driven Testing, which is an approach to describing test cases in a modular way. This standard explains the main concepts and attributes of Keyword-Driven Testing and is applicable to all those who want to create keyword-driven test specifications, create corresponding frameworks, or build test automation based on keywords. This standard defines requirements on frameworks for Keyword-Driven Testing to enable test engineers to share their test artifacts, such as test cases, test data, keywords, or complete test specifications. It also defines minimum requirements for tools supporting Keyword-Driven Testing and defines a common data exchange format to ensure that tools from different vendors can exchange their data (e.g. test cases, test data and test results) [xi]

Software Testing is an internationally agreed set of standards for software testing which can be used by any software development organizations. These standards will help the organization to adopt an internationally-recognized and agreed standards for software testing, which will provide the organization with a high-quality approach to ISO stands for International Standards Organization (recently changed to International Organization for Standardization) and is made up of members representing, for their country, the national body most representative of standardization. ISO, IEC and IEEE and developed standards that can be used by software development organizations throughout the world. These successful implementation of ISO 9126 standard like quality model concept for software process motivates one to develop a similar approach (as its special kind) for the software testing process. Consequently, ISO 9126-like hierarchy of typical evaluation attributes and sub-attributes for any kind software test process will be defined here. The concept of test process reliability is borrowed here from

That of product reliability given in ISO 9126 standard and will be evaluated using product's fault tolerance, recoverability, and predictability. Maturity of the practices is one direct measure of test process reliability.

TMMI

TMMi Foundation, marking the beginning of an industry-wide roadmap for implementing software quality management into the application development lifecycle.

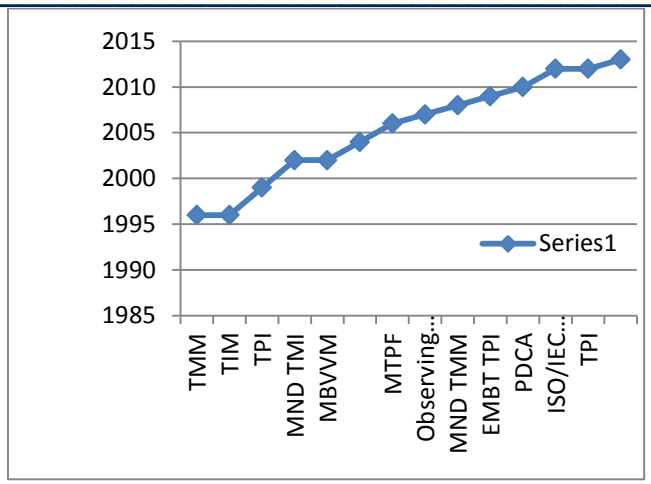
TMMi [xii] is a non-commercial, test maturity model. With TMMi, organizations can have their test processes objectively evaluated by certified assessors, improve their test processes and even have their test process and test organization formally accredited if it complies with the requirements. TMMi uses the concept of maturity levels for process evaluation and improvement. Furthermore process areas, goals and practices are identified. Practical experiences have already shown that applying the TMMi maturity criteria will improve the test process and is likely to have a positive impact on product quality, test productivity, and test lead time.
The full TMMi model (release 1.0) has recently become available and there is rapidly growing world-wide TMMi interest and recognition. As the first version of the TMMi was already published four years ago, many organizations have since used the TMMi to evaluate and improve their test processes. Erik van Veenendaal and Jan Jaap Cannegieter, also co-authors for the "The Little TMMi", have analyzed the results of almost fifty (50) TMMi assessments. The results provide an indication of testing maturity today.

Their roots go back to 2004 when a small group of quality process improvement enthusiasts from around Europe met for the first time and decided it would make sense to develop and support a single, non-commercial test improvement model. Since then, there have been a growing number of supporters who acknowledge the positive difference the TMMi model makes to the delivery of increased quality and reduced costs.

Test Process models and their comparison

The application of software measurement for software engineering evaluations is widely accepted as an effective technique. International standards on process and product quality give pivotal place to measurement. The most well known process maturity model (CMMI) contains a dedicated measurement & analysis process area, while part 3 and 4 Of its counterpart standard for product quality (ISO 9126) also concentrate on product Measurements. Software measurement has successfully been exercised in a variety of evaluation approaches [Duke et al., 2006a], [Ebert et al., 2004] and is seen as one of the critical success factors for process evaluation and improvement [Dyba, 2005].Apart from its unquestionable significance for process, product, and resource evaluations, software measurement has been a key player in all forms of test evaluation approaches discussed in this chapter. All the assessment models of test process place measurement as a requirement in higher maturity levels. For example, it is mentioned as a maturity goal at level 4 in TMM, as a key area in TPI, and as a process area at level 4 in TMMi.

Test Process	Year
1996	TMM
1996	TIM
1999	TPI
2002	MND TMI
2002	MBVVM
2004	TPI automotive
2006	MTPF
2007	Observing Practice
2008	MND TMM
2009	EMBT TPI
2010	PDCA
2012	ISO/IEC 29119/33063
2012	TPI
2013	PDCA evidence based



CONCLUSION

This paper has aimed to bring in the various software quality freameworks and their benefits of each frame work adopted in the software quality management in IT industry which is differentiated from the regular quality management processes followed im any manufacturing industry .

---

[i] IEEE Std 610.12-1990:IEEE standard glossary of software engineering terminology.
[ii] Tassey, G. (2002).

The economic impacts of inadequate infrastructure for software
testing. Technical report, National Institute of Standards & Technology.
[iii] Ericson et al., 1998] Ericson, T., Subotic, A., and Ursing, S. (1998). TIM a test improvement model. J. Softw. Test., Verif. Reliab., 7(4):229–246.

[iv] Jacobs and Trienekens, (2002) JMM : Towards a metric based verification and valuation maturity model. In STEP 2002 Proceedings of the 10th International workshop international on software technology and Engineering Practice, p.123 IEEE Computer Society Press, Washington
[v] Chernak, 2004] Chernak, Y. (2004). Introducing TPAM: Test process assessment model. Crosstalk-The Journal of Defense Software Engineering.
Kollanus S. 2005. Issues in software inspection practices. Proc. of the PROFES 2005 Conference, Oulu, June 13-18. Springer LNCS 3547, 429-442.
[vii] Humphrey, W., (1995) A Discipline for Software Engineering, Addison-Wesley
[viii] Koomen, T. (2002). Worldwide survey on Test Process Improvement. Technical report, Sogeti.
Van Veenendaal, E. and Pol, M. (1997). A test management approach for structured testing. Vchieving Software Product Quality.
[x] Koornen, T.: Worldwide survey on Test. Process Improvement. Technical report, Sogeti (2002)
"ISO/IEC/IEEE 29119-1:2013". *Standards catalogue. International Organization for Standardization. September 2013*. Retrieved 24 June 2018.
[xii] Test Maturity Model integration (TMMi) – Guidelines for Test Process Improvement, (2012), E. van Veenendaal and B. Wells, UTN Publishing (www.utn.nl)
Karuna Devi Mishra and Sudarsanam S.K. Review of literature on Software Testing
Practices and Quality Management approach in Software Industry. *International Journal of Control Theory and Applications, 2016 Vol. 9*, No. 41, pp. 1311-1317.
India ,***Journal of Engineering Computers & Applied Sciences(JECAS)CAS) ISSN No: 2319-5606 Volume 3, No.7, July 2014 Dr.S. K. Sudarsanam, Professor, VIT Business School Chennai, VIT University, Chennai,***
Software Test Process Assessment Methodology – II,Dr.S. K. Sudarsanam, Professor, VIT Business School Chennai, VIT University, Chennai, India
Karuna Devi Mishra and Sudarsanam S.K. IT software Quality Management - A Reverse
Innovation approach for SME's for a Quality Management PracticeReview of literature on Software Testing Practices and Quality Management approach in Software Industry. *International Journal of Control Theory and Applications,2016 Vol. 9*, No. 41, pp. 1303-1310.