What We Know about Software Test Maturity and Test Process Improvement

Garousi, V., Felderer, M., & Hacaloglu, T. (2017). What We Know about Software Test Maturity and Test Process Improvement. *IEEE Software*, *35*(1), 84-92. [8239941]. https://doi.org/10.1109/MS.2017.4541043

Published in:
IEEE Software

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

What we know about software test maturity and test process improvement

Vahid Garousi
Hacettepe University, Ankara, Turkey
Maral Software Engineering Consulting Corporation, Calgary, Canada
vahid.garousi@hacettepe.edu.tr

Michael Felderer
University of Innsbruck, Innsbruck, Austria
michael.felderer@uibk.ac.at

Tuna Hacaloğlu
Middle East Technical University (METU) and Atilim University, Ankara, Turkey
tuna.hacaloglu@metu.edu.tr

Abstract. Software testing practices and processes in many companies are far from being mature and are usually conducted in ad-hoc fashions. Such immature practices lead to various negative outcomes, e.g., ineffectiveness of testing practices in detecting all the defects, and cost and schedule overruns of testing activities. To conduct test maturity assessment (TMA) and test process improvement (TPI) in a systematic manner, various TMA/TPI approaches and frameworks have been proposed. Motivated by a recent industrial project in TMA/TPI and to identify the state-of-the-art and the –practice in this area and to find out what we know about TMA/TPI, we conducted a 'multivocal' literature review (a systematic review from various sources) on both the scientific literature and also practitioners' grey literature (e.g., blog posts) and we present the results in this paper. By summarizing what we know about TMA/TPI, our review identified 58 different test maturity models and a large number of sources with varying degrees of empirical evidence on this topic. This article aims to benefit the readers (both practitioners and researchers) by serving as an evidence-based overview and "index" to the vast body of knowledge in this important and fast-growing area, in assessing and improving the maturity of test processes by benefitting from both the state-of-the-art and –practice.

Keywords. Software testing; test maturity; test process assessment; test process improvement; multivocal literature review; systematic literature review

1 INTRODUCTION

Software testing is an important while a costly phase of the software development life-cycle. A 2013 study by the Cambridge University [1] states that the global cost of detecting and fixing software defects has risen to $312 billion annually and it makes up half of the development time of the average project.

Efficiency and effectiveness of software testing activities and practices vary among different teams and companies. While some companies produce high-quality software by conducting efficient and effective software development and testing activities, unfortunately, according to various studies, e.g., [2-4], software testing practices and processes in many companies are far from being mature and are usually conducted in ad-hoc fashions. Such immature practices lead to various negative outcomes, e.g., ineffectiveness of testing practices in detecting all the defects, and cost and schedule overruns of testing activities.

To assess the quality, efficiency and effectiveness of test activities and practices in a test team or a company, it is common to conduct test maturity assessment (TMA) [5]. As a follow-up, to improve test activities and practices, test process improvement (TPI) is conducted. To conduct TMA and TPI in a systematic manner, various TMA/TPI approaches and frameworks have been proposed in the last several decades. For instance, a 2014 book entitled "*Improving the Test Process: Implementing Improvement and Change*" [5] by the International Software Testing Qualifications Board (ISTQB) suggests various approaches in this context.

Software testing practices and processes are still conducted in ad-hoc fashions in many companies.

In our recent and current collaborations with practitioner testers and in the context of several ongoing TPI projects in which the authors have been involved, we have come to realize that there are several challenges facing a tester/manager interested in conducting TMA/TPI in her/his test processes. Some of the essential aspects and challenges in this context are: raising the need for TMA/TPI among the team members and in the company, planning those activities before actually starting them, identifying the challenges (impediments) beforehand and being ready to address them, and systematic measurement of benefits of TMA/TPI, and finally, to assess the extent to which those activities were successful. To address the above

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

need and to identify the state-of-the-art and –practice in this area and to find out what we (as a community) know about TMA/TPI, we report in this work a 'multivocal' literature review on both the scientific literature and also practitioners' grey literature. A multivocal literature review (MLR) [6, 7] is a systematic literature review (SLR) in which data from multiple types of sources are included, e.g., scientific literature and practitioners' grey literature (e.g., blog posts, white papers, and presentation videos). MLRs have recently started to appear in software engineering, e.g., a recent MLR was published in the area of technical debt [8], and are especially suitable for investigating TMA/TPI which is equally driven by and relevant for both industry and academia.

By summarizing what we know about TMA/TPI, our systematic review identifies 58 different test maturity models and a large number of sources (181) with varying degrees of empirical evidence on this topic. Our article aims to benefit the readers (both practitioners and researchers) in assessing and improving the maturity of test processes by knowing the state-of-the-art and the –practice in this area. 'Review' (survey) papers like this article have previously appeared in the IEEE Software and other venues previously on other topics, e.g., about Agile development [9], and developer motivation [10], and have shown to be useful in providing concise overviews on a given area.

While there exist a few review (survey) papers on the topic of TMA/TPI, e.g., [11, 12], none of the existing surveys have considered both the academic literature and the practitioners' grey literature and also in the depth that we conducted in this study, by identifying 58 test maturity models and also the drivers, challenges and benefits of TMA/TPI.

2 SETTING THE STAGE: GENERAL PROCESS FOR TMA/TPI

Before presenting the results of our review study, we set the stage by discussing briefly the general process for TMA/TPI, which we depict as a UML activity diagram in Figure 1. The idea of this process has been inspired by a simpler version in a book about TMA/TPI [13] and we have extended that process using our findings in this review. Usually, a TMA/TPI initiative starts with need analysis by a test engineer or a team, i.e., determining whether there is enough need to start such an activity. The second step is to obtain awareness among other stakeholders and management. The next step is to determine area(s) of consideration, the TMA/TPI model to be used and the suitable approach. Afterwards, the actual TMA starts which results in identification of TPI areas. TPI is then planned and conducted, and evaluation of TPI outcomes and benefits follows. If there is a need for more TPI, the cycle continues and if not, the process finishes. As we can see in this process, choosing the right models and assessing drivers, challenges and benefits play a major role in conducting TMA/TPI, and our review focuses on latest developments in theses aspects.

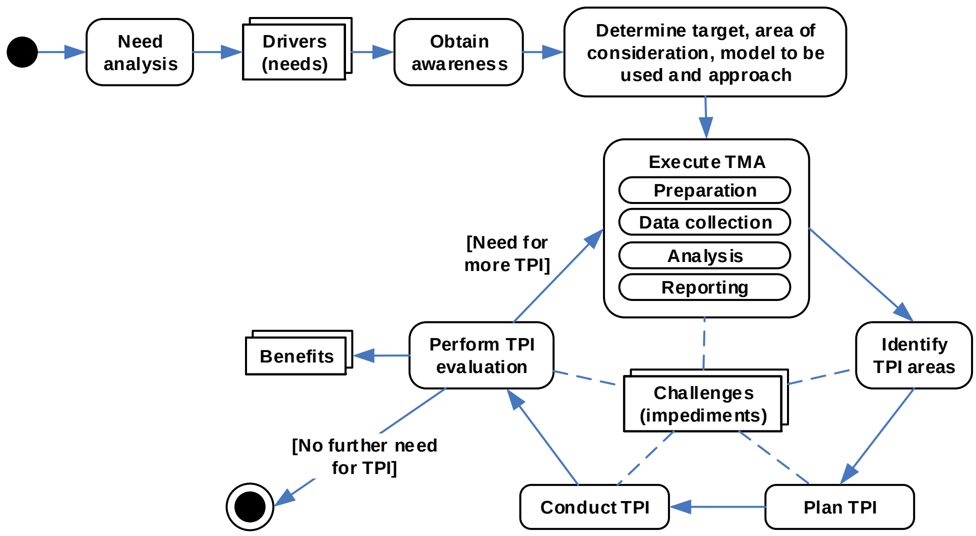

Figure 1- General process for TMA/TPI

3 THE REVIEW PROCEDURE

In our MLR, we followed the standard process for performing SLRs in software engineering. The literature review aimed at addressing the following Review Questions (RQs):

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

- RQ 1-test maturity models: What test maturity models have been proposed in the community by researchers and practitioners?
- RQ 2-drivers: What are the drivers (needs) for TMA/TPI?
- RQ 3-challenges: What are the impediments (challenges) for TMA/TPI?
- RQ 4-benefits: What are the benefits of TMA/TPI?

We performed the searches in the Google and Google Scholar databases. The three authors conducted all the steps as a team. Our search strings were:

- software test maturity
- software test capability
- software test process improvement
- software test process enhancement

To synthesize the opinions and empirical evidence in the primary sources w.r.t. drivers, challenges and benefits, we utilized qualitative coding (also called grounded theory). A more detailed description of our MLR process and qualitative coding is provided in the Web Extras of the article (https://goo.gl/pNCKpn). In addition, we discuss in the 'Web Extras' how we identified and addressed the potential threats to validity of our review.

4 Pool of sources: both the academic community and industry are active

After voting by the authors and applying inclusion/exclusion criteria, our final pool of sources included a total of 181 sources, from which 130 were formally published sources (e.g., conference and journal papers) and 51 were sources in the grey literature (e.g., internet articles and white papers). The final pool of sources and the online mapping repository can be found in an online Google spreadsheet (https://goo.gl/lG4LqF).

In Figure 2, we show (as a stack chart) the plot of number of studies published solely by academic researchers, only by practitioners, or as collaborative works. As we can see, the attention level in this topic has steadily risen since early 1990's by both the research and practitioner communities. Note that the pool of sources for the year 2015 is partial (contains only 5 sources) since the source selection of the study was conducted in June 2015.

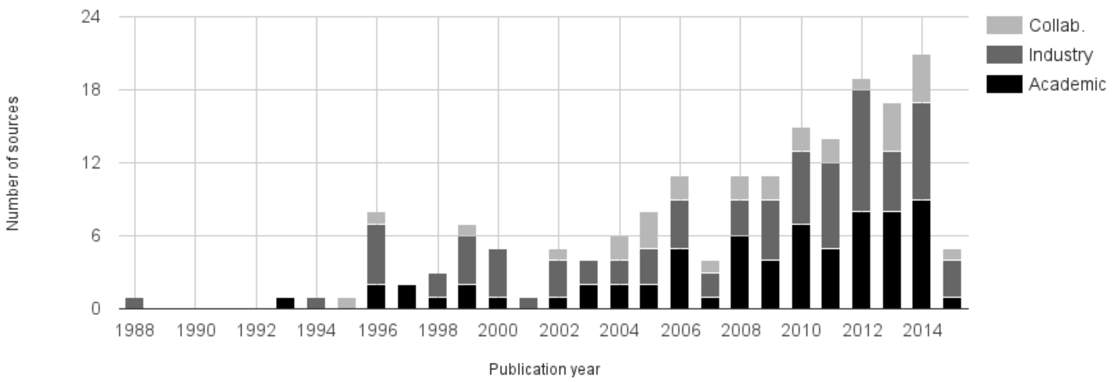

Figure 2-Growth of the TMA/TPI field and authors affiliations of the sources

5 Various test maturity models

Our first review question was to get an idea about the types and characteristics of the existing test maturity models. We differentiated when a given source proposed a model for the first time (either original or extended) and when it used an existing model. 58 of the 181 sources presented new (original or extended) test maturity models while 117 used the existing models for TMA/TPI purposes. The remaining six sources did not propose any new models nor used any of the existing ones, but rather presented other types of contributions for TMA/TPI, e.g., [S 49] proposed a set of test metrics for test strategy evaluation. Being able to see 58 test maturity models out there was quite surprising to us. We are not able to list all of the models in this article, but only present nine representative examples of the 58 models in Table 1, while the full list can be found in the online spreadsheet (https://goo.gl/lG4LqF). The first row shows the three main generic TMA/TPI models, i.e., TMMi, TPI and TestSPICE. The second row shows three TMA/TPI models for specific software development

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

types, i.e., AQAM and ATMM for agile development as well as TPI-EI for embedded software development. Finally, the third row shows TMA/TPI models for specific purposes, i.e., for unit testing (UTMM), test automation (ASTMM) and personal improvement in testing (PTMM). We also mention the levels of the 'staged' TMA/TPI models in Table 1.

In terms of popularity, TMMi (and its earlier version TMM) [#S127] and TPI (and its successor TPI-Next) [#S74] are the most popular models (note that 'S' citations refer to source numbers in the study's online spreadsheet). TMMi and TMM have been used for assessments or as base models in 57 sources while TPI and TPI-Next have been used in 18 sources. 28 sources used other models for TMA/TPI, e.g., TestSPICE [#S93], TMap [#S157].

We are observing the development of models such as TPI-EI [#S24] which is the adoption of the TPI model in the embedded software domain, the Unit Test Maturity Model [#S156], or the Personal Test Maturity Matrix [#S151] which is used to gauge test engineers' (knowledge/skill) maturity and capability development. After reviewing the technical details of several models, authors observed that clearly many aspects in various models overlap.

Similar to the two types of CMMI representations ('staged' vs. 'continuous') [14], the testing maturity models are also, broadly speaking, fall under either of these two types. For instance, TMMi, AQAM and ASTMM (as shown in Table 1) are staged-based models, in which the ranking of levels of conducted on a large set of specific goals and specific practices and a single level is the result of the assessment. On the other hand, models such as TPI, TestSPICE and Personal Test Maturity Matrix are continuous maturity models in which a set of individual key process areas (KPAs) are assessed w.r.t. a set of defined criteria and are given the corresponding levels individually.

What becomes evident from the large set of 58 test maturity models available in the community is that there is no one-size-fits-all model that would fit to all the test improvement needs in industry. Another possible reason for the creation of a subset of the models originating from the academia seems to be that they have not been based on real industrial needs, but rather on hypothetically-argued motivations and also often by not fully reviewing state-of-practice and state-of-the-art to minimize overlaps and to take best practices from research and industry into account.

It is quite surprising to see that more than 58 different test maturity models exist, and many of them have overlapping aspects.

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

Table 1-Examples of the test maturity models proposed in the community along with their maturity levels

<table>
<tr>
<td>Test Maturity Model integration (TMMi) [#S127]
• Level 1: Initial
• Level 2: Definition
• Level 3: Integration
• Level 4: Management and measurement
• Level 5: Optimization</td>
<td>TPI (Test process improvement) [#S74]

A 'continuous' model, i.e., not 'staged' (based on maturity levels), but including 20 Key Performance Areas (KPAs). Each KPA has four levels: A…D
1. Test strategy
2. Life-cycle model
3. Moment of involvement
4. Estimating and planning
….
18. Test process management
19. Evaluation
20. Low-level testing</td>
<td>TestSPICE [S93]

A set of KPAs. Based on ISO/IEC 15504, Software Process Improvement and Capability dEtermination (SPICE) standard</td>
</tr>
<tr>
<td>Agile Quality Assurance Model (AQAM) [#S3]
• Level 1: Initial
• Level 2: Performed
• Level 3: Managed
• Level 4: Optimized</td>
<td>Agile Testing Maturity Model (ATMM) [S35]
• Level 0: Waterfall
• Level 1: Forming
• Level 2: Agile bonding
• Level 3: Performing
• Level 4: Scaling</td>
<td>TPI-EI [S24]

Adaptation of TPI for embedded software</td>
</tr>
<tr>
<td>Unit Test Maturity Model (UTMM) [S156]
• Level 0: Ignorance
• Level 1: Few simple tests
• Level 2: Mocks and stubs
• Level 3: Design for testability
• Level 4: Test driven development
• Level 5: Code coverage
• Level 6: Unit tests in the Build
• Level 7: Code coverage feedback Loop
• Level 8: Automated builds and tasks</td>
<td>Automated Software Testing Maturity Model (ASTMM) [S5]
• Level 1: Accidental automation
• Level 2: Beginning automation
• Level 3: Intentional automation
• Level 4: Advanced automation</td>
<td>Personal Test Maturity Matrix (PTMM) [#S151]

A set of KPAs such as: test execution, automated test support and reviewing</td>
</tr>
</table>

To visually demonstrate the growth of TMA/TPI models over time, we depict in Figure 3 the chronological evolution graph of TMA/TPI models and their relationships, i.e., how models build on top of (based on) earlier models. The idea for this figure has been inspired by a similar evolution model prepared for the UML[1]. As we can see, new (original or extended) TMA/TPI models have been proposed since 1985 in a regular pace. Many of the new models are based on older models, e.g., MB-TMMi (MB standing for metric-based) [Source 48], proposed in 2001, is based on TMM [Source 44].

[1] https://en.wikipedia.org/wiki/Unified_Modeling_Language#/media/File:OO_Modeling_languages_history.jpg

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

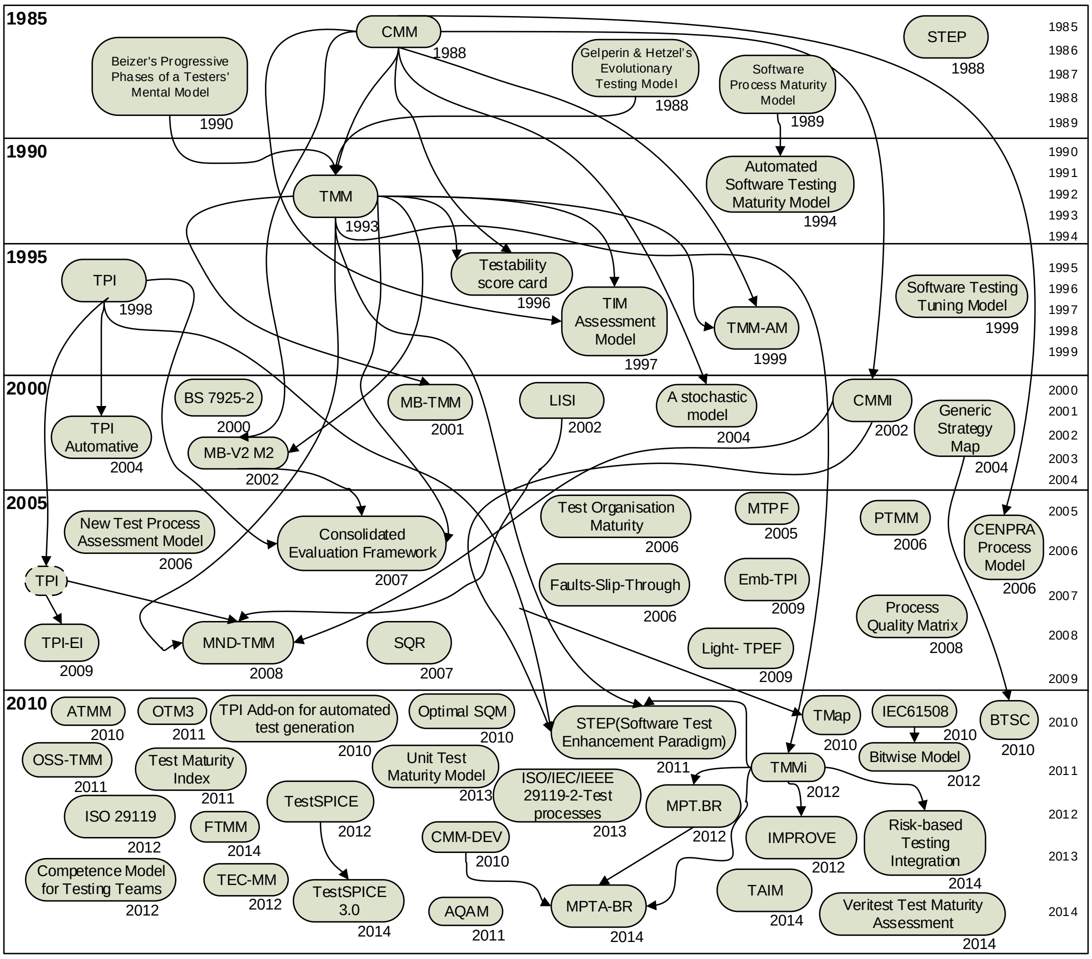

Figure 3-Evolution of TMA/TPI models and their relationships

With such a large collection of models and the overlap among them, when a test team or a manager decides to conduct TMA/TPI activities, it would not be easy to choose the most suitable model(s) to apply, a challenge reported in the previous work [15] and also experienced by the authors in their ongoing industrial projects, e.g., [16]. To further add to the complexity of conducting TMA/TPI using these models, many practitioners and researchers have reported challenges when using even established models such as the TMMi [15], e.g., not being able to objectively assess each maturity area/item using the existing model guidelines.

6 Drivers (needs) for TMA/TPI

Similar to other types of assessment or improvement activities, to start TMA/TPI activities in a team, unit or organization, there should be enough drivers (needs) to justify the energy/time and money to be spent on TMA/TPI activities. After careful data extraction, logging of drivers phrased in different forms and terminologies, and qualitative coding of the drivers as reported in the sources, we synthesized and classified drivers (needs) into five categories: process and operational needs (mentioned in 46 sources), needs related to software quality (25 sources), cost-related needs (23 sources), time and schedule-related needs (12 sources), and 'other' needs (15 sources).

Concrete examples of the process and operational needs, as mentioned in the sources, are as follows: lack of focus in test activities and people-dependent performance [#S23], low test efficiency [#S56], not meeting expectations or commitments [#S58], internal stakeholder dissatisfaction [#S58], missing exit criteria for testing [#S70], need for improving the productivity [#S73], raising profile of testing, baseline test capabilities, and developing credible testing roadmap [#S159].

Drivers for TM/TPI fall into five categories: process and operational needs, needs related to software quality, cost-related needs, time and schedule-related needs.

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

Examples of needs related to software quality, as mentioned in the sources, are as follows: high number of faults due to low testing quality [#S4], direct relationship between the quality of the test process to the final quality of the developed product [#S29], and lack of planning and resources for testing impacting software quality [#S40].

Examples of cost-related needs are: the argument that most current testing processes are often technique-centered, rather than organized to maximize business value [#S78], testing costs being too high [#S177], and low cost effectiveness of testing [#S181].

Examples of schedule-related needs are: delays in production due to ineffective testing [#S4], accelerating time-to-market by effective testing [#S25], and test team spending a lot of time on manual testing [#S28].

Many sources report that one of the main steps in starting (and the success of) TMA/TPI activities are to get (and keep) stakeholders' commitment. To establish commitment, an important activity is cost-benefit analysis (both quantitative and qualitative) for these activities. Costs in this context relate to the efforts to be spent on the activities and benefits relate to satisfying drivers and needs. Only if the expected benefits outweigh the costs, TMA/TPI activities will get the green light to start.

7 Challenges (impediments) for TMA/TPI

Any improvement activity will come with its own challenges (impediments). After careful data extraction, we classified the challenges into eight categories: lack of (required) resources (mentioned in 17 sources), lack of competencies (in 7 sources), resistance to change (12 sources), improving feels like "an additional effort" (9 sources), no clear benefits seen (4 sources), unclear scope and focus (7 sources), no owner for the improvement (5 sources), and "other" challenges (23 sources).

For example, [#S23] discussed the lack of a process improvement infrastructure at the Union Switch and Signal (US&S), a supplier of railway signaling equipment in the US, as a major barrier for TPI in that company. In [#S123], which was a paper titled "*Ten factors that impede improvement of verification and validation processes in software intensive organizations*", lack of available human and economic resources was considered an important challenge for small and medium organizations. In terms of lack of competencies, [#S113] recommended that testers should be trained to conduct TPI.

In terms of resistance to change, [#S155] recommended that TMA efforts should be tailored to meet the cultural norms of the organization or there will be resistance. The authors of [#S89] focused on personal psychology of testers and report that, by minimizing the fear factor of applying TPI, they put testers through fewer emotional swings in the course of TPI.

Test process improvement often feels like "an additional effort".

As a result of an empirical study, [#S123] reported that the activities dedicated to diagnosing the current practice, in relation to the TPI activities, are often felt like "an additional effort".

Reported by a team of Brazilian practitioners and researchers, [#S102] reported that small companies aiming to implement TPI models may abort this undertaking. This may be because the models did not by then show benefits or because the company is not ready for the maturity improvement. As another example, [#S123] reported that it is often very difficult to estimate the expected return on investment (ROI) of TPI activities. Moreover, such estimations usually have a low degree of reliability.

In terms of unclear scope and focus, [#S62] mentioned that a major challenge is to prioritize the areas to know where to focus the improvement activities. Without such a decision support, it is common that improvements are not implemented because organizations find them difficult to prioritize.

8 Benefits of TMA/TPI

The successful implementation of TMA/TPI heavily depends on expected or actual benefits for a team, unit or organization. After careful data extraction, we classified the benefits into three categories: business (economic) benefits (mentioned in 27 sources), operational benefits (48 sources), as well as technical benefits (37 sources).

Examples for business (economic) benefits of TMA/TPI are increased profit [#S78], increased customer satisfaction [#S78], positive return on investment [#S55, #S92], reduced cost of test tasks [#S66], reduction of defect costs [#S112],

Benefits of TM/TPI fall into three categories: business (economic) benefits, operational benefits, and technical benefits.

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

better internal and external reputation [#S146], increased business opportunities [#S146], and reduced support cost [#S180].

Examples for operational benefits of TMA/TPI are shorter development time [#S121], lower development cost [#S121], better planning of testing costs [#S71], alignment of internal testing processes with external value objectives [#S78], better adherence to release dates [#S79], reduced failure administration [#S59], minimized test cycle time [#S96], more effective and better risk identification and management [#S146], development of adequate training for test personnel [#S43], and process control based on metrics [#S170] resulting in more accurate estimations and predictions [#S11].

Examples for technical benefits are a reduced number of field defects resulting in better software quality in general [#S79], reduction of high severity defects [#S180], increased traceability to support release decisions [#S88], improved test automation [#S117], as well as improved test design by adoption of new techniques [#S133].

9 An industrial case-study

As discussed earlier, this review study was partly motivated by a recent industrial project in the scope of TMA/TPI and we wanted to find out the existing models and approaches out there as to be able to choose the most suitable model for that project.

The company under study is a Turkish software firm and one of the industry partners of the first author. The company had an interest to increase maturity of its test practices. This review study helped us identify the 58 test maturity models and the related drivers, challenges and benefits, and helped us plan and conduct the TMA/TPI project in a more rigorous and systematic manner. Assessments using TMMi were conducted using TMMi specification (reference model), release 1.0 [17], and the TMMI Assessment Method Application Requirements (TAMAR) document [18]. TMMI has five maturity levels: (1) Initial, (2) Managed, (3) Defined, (4) Measured, and (5) Optimization. Each maturity level has several process areas (PA). Each PA has several specific goals (SG) and specific practices (SP). There are in total 50 specific goals (SG) and 188 specific practices (SP). For instance, under the maturity level 2 (managed), there are five process areas, e.g., PA 2.1 (test policy and strategy). This PA has three SGs: SG 1 (establish a test policy), SG 2 (establish a test strategy), and SG 3 (establish test performance indicators). The above SG 1, in turn, has three SPs: SP 1.1 (define test goals), SP 1.2 (define test policy), and SP 1.3 (distribute the test policy to stakeholders).

To conduct the TMMi assessment, we used the five-point scale for ratings each SP suggested by the TAMAR document [18]: fully achieved (FI, "I" for implemented), largely achieved (LI), partially achieved (PI), not achieved (NI), and not applicable (N/A), which is similar to the Standard CMMI Appraisal Method for Process Improvement (SCAMPI)'s five levels: fully implemented (FI), largely implemented (LI), partially implemented (PI), not implemented (NI), and not yet (NY).

Once the TMMi assessment was conducted in the company under study, to systematically evaluate the test maturity and to compile the areas for improvement, we reviewed and collected the areas (the so-called "specific practices" in TMMi) which were ranked lower than 'fully achieved' (FI). Here are several of those areas: (1) Separation of debugging from testing can be more clear in test policy documents, (2) It is a good idea to document generic product risks in potential risks documents, (3) Test policy and test performance indicators and metrics could be updated, and (4) There is a need for more systematic risk-based testing. Thanks to the results of this review, we knew the potential drivers, challenges and benefits before starting the project and interesting noticed several of them throughout our activities. In fact, many of the team members in the company are happy that this review helped them to be prepared for the project. The TPI activities have now been planned and are currently underway in the company under study.

10 Summary of findings and road ahead

In this multivocal literature review (MLR) based on scientific literature as well as practitioners' grey literature (e.g., internet articles and white papers), we summarized and synthesized the body of knowledge on test maturity and test process improvement. Readers of this article (both practitioners and researchers) will benefit from the provided evidence-based overview and "index" to the comprehensive body of knowledge in the fast-growing area of TMA/TPI which is equally important for industry and academia. Structured and scoped by four Review Questions (RQs), the review was able to systematically group and synthesize all the existing TMA/TPI models, drivers (needs) for TMA/TPI, the impediments (challenges), and benefits of TMA/TPI.

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

The growing number of scientific papers shows that TMA/TPI is an active area of research. Furthermore, the high number of online articles by practitioners as well as the high number of reported test maturity models highlights the practical relevance of the topic. In terms of popularity, TMMi (and its earlier version TMM) and TPI are the most popular models. Main drivers for TMA/TPI are related to process and operational needs, needs related to software quality, cost-related needs, as well as time and schedule-related needs. Main challenges for TMA/TPI are related to lack of (required) resources, lack of competencies, resistance to change, improving feels like "an additional effort", no clear benefits seen, unclear scope and focus, as well as no owner for the improvement.

For a reader interested in conducting TMA/TPI in her/his test processes, some of the essential aspects are: (1) raising the need for TMA/TPI among the team members and in the company, (2) choosing the "right" maturity model(s) to apply, (3) planning those activities before actually starting them, (4) identifying the challenges (impediments) beforehand and being ready to address them, and (5) systematic measurement of benefits of TMA/TPI to assess the extent to which those activities were actually successful. The issue of choosing the right maturity models to apply has to some extent been explored in other areas, e.g., in business process maturity assessment [19], and a topic which needs further investigation for the set of test maturity models. For a practitioner new to this area who intends to conduct TMA/TPI, it is important to know the differences and similarities among the models and also success/failure of the models. There is a need for domain analysis on the models and in-depth examinations of the extent to which the maturity models are similar and tend to become unified. Also, another important issue is assessing the models' "fit for purpose", i.e., to what extent do they really help test teams to assess and improve their test processes? This issue has also been slightly explored in the domain of business process maturity assessment [20] and requires investigation in the scope of test maturity models as well.

Although there has been a high interest and progress in the areas of TMA/TPI, there is need for more empirical studies providing evidence for TMA/TPI in specific contexts, e.g., by taking the domain of the systems under test into account, and need for more evidence-based approaches in this area.

REFERENCES

[1] T. Britton, L. Jeng, G. Carver, P. Cheak, and T. Katzenellenbogen, "Reversible Debugging Software," *University of Cambridge, Judge Business School, Tehnical Report*, 2013.

[2] V. Garousi, A. Coşkunçay, A. B. Can, and O. Demirörs, "A Survey of Software Engineering Practices in Turkey," *Journal of Systems and Software*, vol. 108, pp. 148-177, 2015.

[3] V. Garousi and J. Zhi, "A Survey of Software Testing Practices in Canada," *Journal of Systems and Software*, vol. 86, pp. 1354-1376, 2013.

[4] M. Grindal, J. Offutt, and J. Mellin, "On the testing maturity of software producing organizations," in *Testing: Academia & Industry Conference-Practice And Research Techniques*, 2006.

[5] G. Bath and E. V. Veenendaal, *Improving the Test Process: Implementing Improvement and Change - A Study Guide for the ISTQB Expert Level Module*: Rocky Nook, 2013.

[6] R. T. Ogawa and B. Malen, "Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method," *Review of Educational Research*, vol. 61, pp. 265-286, 1991.

[7] V. Garousi, M. Felderer, and M. V. Mäntylä, "The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature," in *International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2016, pp. 171-176.

[8] E. Tom, A. Aurum, and R. Vidgen, "An exploration of technical debt," *Journal of Systems and Software*, vol. 86, pp. 1498-1516, 6// 2013.

[9] T. Dyba and T. Dingsoyr, "What Do We Know about Agile Software Development?," *Software, IEEE*, vol. 26, pp. 6-9, 2009.

[10] T. Hall, H. Sharp, S. Beecham, N. Baddoo, and H. Robinson, "What Do We Know about Developer Motivation?," *Software, IEEE*, vol. 25, pp. 92-94, 2008.

[11] W. Afzal, S. Alone, K. Glocksien, and R. Torkar, "Software test process improvement approaches: A systematic literature review and an industrial case study," *Journal of Systems and Software*, vol. 111, pp. 1-33, 1// 2016.

[12] C. Garcia, A. Dávila, and M. Pessoa, "Test Process Models: Systematic Literature Review," in *Software Process Improvement and Capability Determination*. vol. 477, A. Mitasiunas, T. Rout, R. O'Connor, and A. Dorling, Eds., ed: Springer International Publishing, 2014, pp. 84-93.

[13] T. Koomen and M. Pol, *Test Process Improvement: A Practical Step-by-step Guide to Structured Testing*: Addison-Wesley, 1999.

[14] S. Cepeda, "CMMI - Staged or Continuous?," *https://www.sei.cmu.edu/library/assets/cepeda-cmmi.pdf*, 2005, Last accessed: Nov. 2015.

This is the pre-print of a paper that has been published with the following DOI:
http://dx.doi.org/10.1109/MS.2017.4541043

[15] K. Rungi and R. Matulevičius, "Empirical Analysis of the Test Maturity Model Integration (TMMi)," in *Information and Software Technologies*. vol. 403, T. Skersys, R. Butleris, and R. Butkiene, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 376-391.

[16] M. Felderer and R. Ramler, "Integrating risk-based testing in industrial test processes," *Software Quality Journal*, vol. 22, pp. 543-575, 2014.

[17] TMMI Foundation, "TMMI specification (reference model), release 1.0," *http://www.tmmi.org/pdf/TMMi.Framework.pdf*, Last accessed: Oct. 2015.

[18] TMMI Foundation, "TMMi Assessment Method Application Requirements (TAMAR), Version 2.0," *http://www.tmmi.org/sites/all/themes/softtech/pdf/TMMi.TAMAR.pdf*, Last accessed: Oct. 2015.

[19] A. Van Looy, M. De Backer, G. Poels, and M. Snoeck, "Choosing the right business process maturity model," *Information & Management*, vol. 50, pp. 466-488, 11// 2013.

[20] A. Van Looy, "Looking for a Fit for Purpose: Business Process Maturity Models from a User's Perspective," in *IFIP Working Conference on Enterprise Information Systems of the Future*, 2013, pp. 182-189.

AUTHOR BIOGRAPHIES

Vahid Garousi is an Associate Professor of Software Engineering in Hacettepe University in Ankara, Turkey, and the Managing Director of Maral Software Engineering Consulting Corporation, based in Calgary, Canada. Prior to that, he was a tenured Associate Professor at the University of Calgary, Canada from 2006 until 2014. His research interests include software test engineering, software quality, empirical software engineering and improving industry-academia collaborations in software engineering. He was selected a Distinguished Visitor (speaker) for the IEEE Computer Society's Distinguished Visitors Program (DVP) during the period of 2012-2015. Contact him at vahid.garousi@hacettepe.edu.tr

Michael Felderer is a senior researcher at the University of Innsbruck, Austria. He holds a PhD and habilitation degree in computer science. His research interests include software and security testing, software processes, requirements engineering, empirical software engineering, as well as improving industry-academia collaboration. Besides his research activities, he transfers his research results into practice as a consultant and speaker on industrial conferences. Contact him at michael.felderer@uibk.ac.at

Tuna Hacaloğlu is a PhD student in the Department of Information Systems in Informatics Institute of the Middle East Technical University, Ankara, Turkey where she received her masters degree in 2013. She works as an instructor in the Department of Information Systems Engineering of Atilim University. Her research interests include information systems, software engineering, software engineering education, and software testing. Contact her at tuna.hacaloglu@metu.edu.tr