Information and Software Technology 000 (2017) 1–27

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Software test maturity assessment and test process improvement: A multivocal literature review

Vahid Garousi[a,b,*], Michael Felderer[c], Tuna Hacaloğlu[d,e]

[a] Software Engineering Research Group, Department of Computer Engineering, Hacettepe University, Ankara, Turkey
[b] Maral Software Engineering Consulting, Corporation, Calgary, Canada
[c] Quality Engineering Research Group, Institute of Computer Science, University of Innsbruck, Innsbruck, Austria
[d] Informatics Institute, Middle East Technical University (METU), Ankara, Turkey
[e] Department of Information Systems Engineering, Atilim University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 7 August 2016
Revised 23 November 2016
Accepted 4 January 2017
Available online xxx

Keywords:
Software testing
Test management
Test process
Test maturity
Test process assessment
Test process improvement
Multivocal literature review
Systematic literature review

ABSTRACT

Context: 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 models and approaches have been proposed.

Objective: It is important to identify the state-of-the-art and the –practice in this area to consolidate the list of all various test maturity models proposed by practitioners and researchers, the drivers of TMA/TPI, the associated challenges and the benefits and results of TMA/TPI. Our article aims to benefit the readers (both practitioners and researchers) by providing the most comprehensive survey of the area, to this date, in assessing and improving the maturity of test processes.

Method: To achieve the above objective, we have performed a Multivocal Literature Review (MLR) study to find out what we know about TMA/TPI. A MLR is a form of a Systematic Literature Review (SLR) which includes the grey literature (e.g., blog posts and white papers) in addition to the published (formal) literature (e.g., journal and conference papers). We searched the academic literature using the Google Scholar and the grey literature using the regular Google search engine.

Results: Our MLR and its results are based on 181 sources, 51 (29%) of which were grey literature and 130 (71%) were formally published sources. 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. We also conducted qualitative analysis (coding) to synthesize the drivers, challenges and benefits of TMA/TPI from the primary sources.

Conclusion: We show that current maturity models and techniques in TMA/TPI provides reasonable advice for industry and the research community. We suggest directions for follow-up work, e.g., using the findings of this MLR in industry-academia collaborative projects and empirical evaluation of models and techniques in the area of TMA/TPI as reported in this article.

1. Introduction

Software testing is an impotent 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.

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. Also, testing is often conduct not efficiently, e.g., "*The costs of testing of a software project or product are considerable and therefore it is important to identify process improvement propositions for testing*" [5].

To conduct Test Maturity Assessment (TMA) and Test Process Improvement (TPI), together referred to as TMA/TPI, in a systematic manner, various TMA/TPI approaches and frameworks have been

* Corresponding author.
E-mail addresses: vahid.garousi@hacettepe.edu.tr (V. Garousi), michael.felderer@uibk.ac.at (M. Felderer), tuna.hacaloglu@metu.edu.tr (T. Hacaloğlu).

http://dx.doi.org/10.1016/j.infsof.2017.01.001
0950-5849/

proposed. For example, a 2014 book entitled "*Improving the Test Process: Implementing Improvement and Change*" [6] by the International Software Testing Qualifications Board (ISTQB) suggest various approaches in this context.

To identify the state-of-the-art and the –practice in this area of scientific and practical interest and to find out what we know about TMA/TPI, we report in this work a 'multivocal' literature review on both the scientific literature and also practitioners' grey literature and we present its results in this article. A multivocal literature review (MLR) [7–9] is a type of systematic literature reviews (SLR) in which data from various sources are included, e.g., scientific literature and practitioners' grey literature (e.g., blog posts, white papers, and presentation videos). We believe that conducting a MLR in the area of TMA/TPI will be more useful compared to a SLR since there is a large of body of knowledge and experience reported by practitioners in the grey literature (e.g., blog posts and white papers) which a regular SLR study will not review and synthesize (by being limited to only the formal published literature).

MLRs have recently started to appear in software engineering, e.g., in recent ones in the areas of technical debt [7] and test automation [10], respectively. Furthermore, the need for more MLRs in software engineering has recently been pointed out and investigated empirically [11], especially also by pointing to the field of test process improvement, which is of high interest to research and practice. By summarizing what we know about TMA/TPI, our systematic review identified 58 different test maturity models and a large number of sources 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 benefitting the state-of-the-art and the –practice in this area.

While there exist a few review (survey) papers on the topic of TMA/TPI, e.g. [12, 13], none of the existing surveys have considered both the academic literature and the practitioners' grey literature and also in the depth that we have conducted in this study, by identifying 58 test maturity models and also the drivers, challenges and benefits of TMA/TPI.

On another note, we would like to clearly note the scope of this study before continuing with the rest of the study. We are aware that testing is not the only approach for software quality assurance, verification and validation (V&V). Techniques such as formal methods, inspections, static code analysis and peer reviews as other forms of V&V that are complementary to testing. But to keep our MLR study focused, we have only focused on surveying the maturity assessment and process improvement approaches specific to software testing and have excluded those focusing on the other V&V activities, e.g., studies such as [14, 15]. Certainly, we encourage MLR and other types of review studies on those other subareas of software quality assurance.

The remainder of this article is structured as follows. A review of the related work is presented in Section 2. We describe the study goal and research methodology in Section 3. Section 4 presents the searching phase and selection of sources. Section 5 discusses the development of the systematic map and data-extraction plan. Section 6 presents the results of the MLR. Section 7 summarizes the findings and discusses the lessons learned. Finally, in Section 8, we draw conclusions, and suggest areas for further research.

2. Background and related work

In this section, we first provide a brief overview of the technical domain of this MLR (software test maturity assessment and test process improvement). We then briefly provide a background on multivocal literature reviews (MLRs) since it is a relatively new terminology in SE. We finish the section by reviewing the related work, i.e., other secondary studies in the scope of TMA/TPI and how our work differs from them and contributes new knowledge to the literature.

2.1. Brief overview of software test maturity assessment and test process improvement

Software testing practices and processes in many companies are far from mature and are still usually conducted in ad-hoc fashions [2–4]. Thus, many team and companies are interested to assess and improve the maturity of their software testing practices and processes.

A recent 2016 SLR [12] on this topic identified 18 TPI approaches showing the fast progress of this important field in software testing. According to many sources (e.g., [12]), TMMi [16, 17] and TPI [18] (and its newer version TPI-Next [19]) are the most popular and widely-used models and approaches in this area. We provide a brief overview of TMMi in the following.

TMMi is based on TMM itself, which in turn is based on the Capability Maturity Model (CMM) and CMMI, and was first proposed in 1998 [20]. The latest version of TMMi specification as of this writing is 1.0 [17] which is prepared and published by the TMMi Foundation in 2012.

Fig. 1 shows TMMi maturity levels and process areas and Fig. 2 shows its structure and components. As the structure outlines, each maturity level has several process areas (PA), and each process area has several specific goals and specific practices. In total, under the four maturity levels (2, 3 and 4), the TMMi [17] specified 50specific goals (SG) and 188 specific practices (SP). For example, under the 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.

In this context, it is also important to discuss the general process for TMA/TPI. In a 1999 book, Koomen and Pol nicely summarize that process as shown in Fig. 3, which starts with obtaining awareness, i.e., pinpointing the need for TMA/TPI.

2.2. Multivocal literature reviews

While SLR and SM studies are valuable, researchers have reported that "*the results of a SLR or a SM study could provide an established body of knowledge, focusing only on research contributions*" [21]. Since these secondary studies do not include the "grey" literature (non-published, nor peer-reviewed sources of information), produced constantly in a very large scale by practitioners, those studies do not provide much insight into the "state of the practice". For a practical (practitioner-oriented) field such as SE, synthesizing and combing both the state-of-the art and –practice is very important. Unfortunately, it is a reality that a large majority of software practitioners do not publish in academic forums [22], and this means that the voice of the practitioners is limited if we do not consider grey literature in addition to academic literature in review studies.

2.2.1. MLRs in other fields

The term Multivocal Literature Review (MLR) was defined in the early 1990 s in other fields, e.g., in educational research [8], as SLR which includes both the academic (formal) and the grey (informal) literature. The main difference between an MLR and a SLR or a SM is the fact that, while SLRs and SMs use as input only academic peer-reviewed articles, in MLRs, grey literature, such as blogs, white papers and web-pages, is also considered as input

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

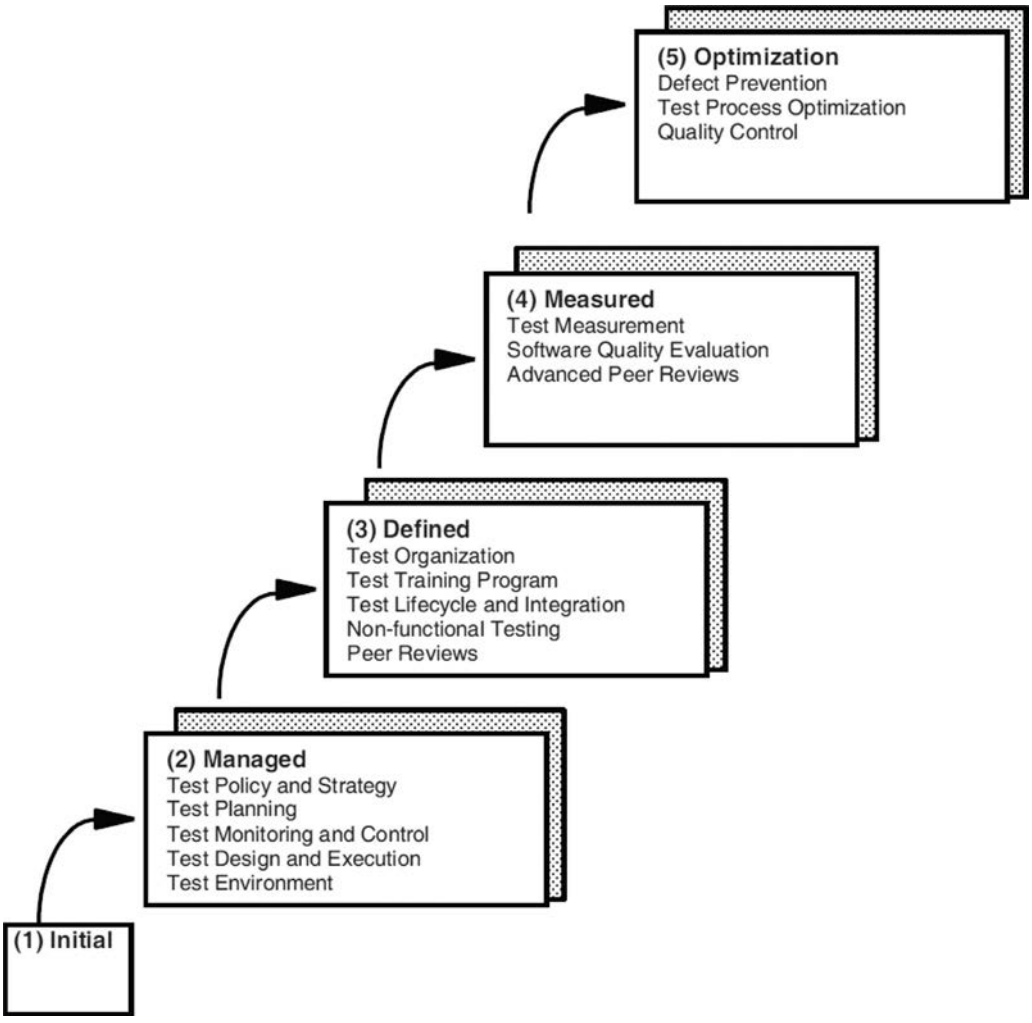

Fig. 1. TMMi maturity levels and process areas (taken from [17]).

[21]. A multivocal synthesis is suggested as an appropriate tool for investigations in a field "*characterized by an abundance of diverse documents and a scarcity of systematic investigations*" [23]. Researchers also believe that: *"another potential use of multivocal literature reviews is in closing the gap between academic research and professional practice*" [8].

While the notions of "MLR" and "multivocal" have been used in the community, still many sources use the "grey" literature terminology and whether/how to include them in SLRs, e.g., [24–26]. For example, [24] discusses the advantages and challenges of including grey literature in state-of-the-evidence reviews, in the context of evidence-based nursing. [25] discusses the challenges and benefits of searching for grey literature in SLRs.

A 1991 paper [8] discussed rigor in MLRs and proposed a method based on exploratory case study to conduct MLRs rigorously. Hopewell et al. [27] conducted a review of five studies, in the area of evidence-based medicine, comparing the effect of the inclusion or exclusion of 'grey' literature in meta-analyses of randomized medical trials. Results of this paper show that trials in formally published literature had more participants on average and that most common types of grey literature in this context were abstracts unpublished data. However, the authors also highlight that there is limited evidence to show whether trials published in grey literature are of poorer methodological quality than formally published trials. The issue of the grey literature is such important that there is even an International Journal on the topic of Grey Literature (www.emeraldinsight.com/toc/ijgl/1/4).

2.2.2. MLRs in sE

The 'multivocal' terminology and the inclusion of grey literature have only been recently started to appear in the SLRs in SE, i.e., since 2013 in [7]. We found only a few SLRs in SE which explicitly used the 'multivocal' terminology: [7, 10, 21, 28]. [7] is a 2013 MLR on 'technical debt'. [21] is a 2015 MLR on the financial aspect of managing technical debt. [28] is a 2015 MLR on iOS applications testing. Finally, [10] is a 2016 MLR to decide when and what to automate in software testing in which the first author of the current paper was involved.

Many other SLRs have included the grey literature in their reviews and have not used the 'multivocal' terminology, e.g., [29]. A 2012 MSc thesis entitled "*On the quality of grey literature and its use in information synthesis during systematic literature reviews*" [30] explored the state of including the grey literature in the SE SLRs. Two of the RQs in that study were: (1) What is the extent of usage of grey literature in SE SLRs?, (2) How can we assess the quality of grey literature? The study found that the ration of grey evidence in the SE SLRs were only about 9.22%, and the grey literature included concentrated mostly in recent past (~48% in last 5 years 2007–2012).

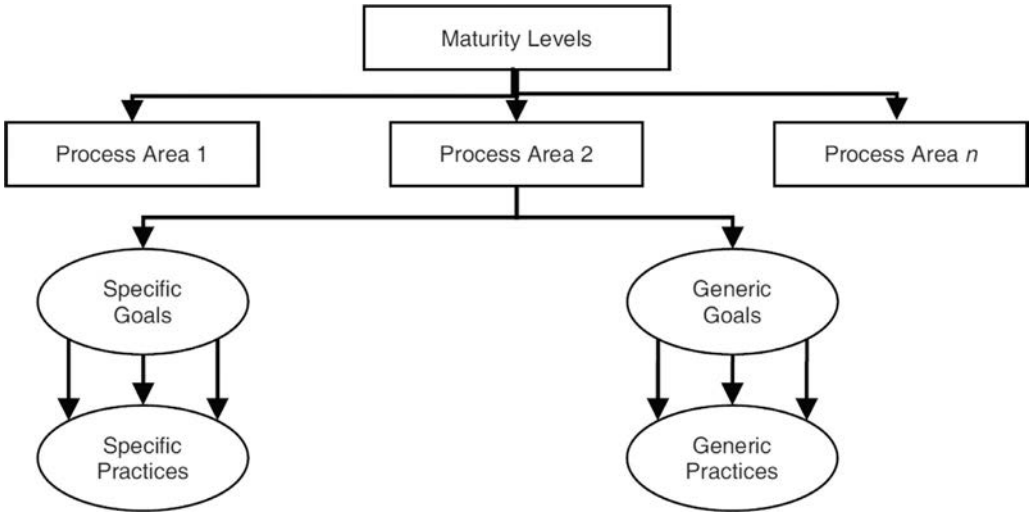

Fig. 2. TMMi structure and components (taken from [17]).

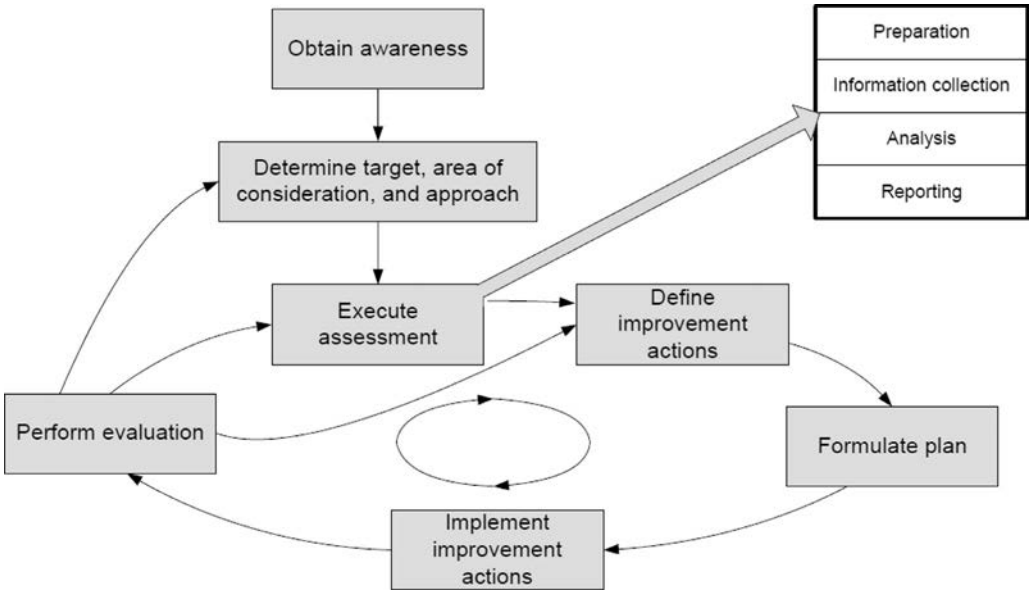

Fig. 3. General process for TMA/TPI (taken from [18]).

A recent study [11] in 2016, in which two of the study authors were involved, pointed out the need for multivocal literature reviews in SE and empirically investigated the issues. Based on several sample SLRs and MLRs in the areas of GUI testing, metrics in agile and lean, test automation, and test process improvement, the study identified the missing and gained knowledge due to excluding or including grey literature in review studies. The authors of [11] found that (1) grey literature can give substantial benefits in certain areas of SE including TMA/TPI, and that (2) the inclusion of grey literature brings forward certain challenges since evidence reported in grey literature is often less rigorous compared to formal literature and mostly based on experience and opinion. Let us note that, when conducting MLRs, one should assess the accuracy or quality of the knowledge shared in the grey literature, which is also done in SLR studies. But neither [11] nor any of the SLR/SM guidelines in SE (e.g., [31–33]) have provided heuristics or hints for (systematic) quality assessment of sources in the grey literature. Other fields have discussed this issue to some extent, e.g., [30, 34, 35], which could be adopted in the SE domain (perhaps after some modifications to make them fit to SE).

2.3. Related works: other secondary studies in the scope of TMA/TPI

While conducting our search for primary sources in this MLR, we also developed a pool of all "related" studies, i.e., other secondary studies (review papers) in the area of TMA/TPI. Table 1 shows a summary on those studies and the relationship of our MLR to them. We divided the related work into two sets: (1) studies in which the literature review is the main goal [12, 13, 36–39], and (2) those in which the review is only a part (sub-goal) [40–44]. The papers in the second category usually had conducted a brief review (survey) as part of their work, e.g., a thesis [41] in which its first part surveyed and summarized existing evaluation approaches. We are also showing the number of primary studies (for secondary studies) and the number of references (for regular studies), and also the number of TMA/TPI models reviewed in each study.

As we can see in Table 1, the number of TMA/TPI models reviewed in various studies is quite low (between 2 and 23) compared to the comprehensive set of models that we have summarized in this review (58 models). We have also included in

Table 1
A mini-SM on other reviews (secondary studies) in TMA/TPI and relationship to our MLR.

	Reference	Year	Title	Research methodology	# of primary studies / references	# of models reviewed	Other contributions
Review is the main goal	[36]	2000	A comparison of TMM and other test process improvement models (a technical report by Frits Philips Institute)	Informal comparison	32	8	Compared the most important TPI models available (up to 2000) and gave input for the development of a new model named Metric-based Test Maturity Model (MB-TMM).
	[37]	2007	Research directions in verification & validation process improvement	Informal survey (review)	26	10	Identified the following potential research directions: need for improvement of Testing Maturity Model (TMM), answering 'how' instead of 'what' aspect of V&V processes, need for 'product-focused' V&V process improvement, and the need for V&V in 'emerging' development environments such as service-oriented V&V and V&V process in small IT organizations
	[38]	2012	Test process improvement - evaluation of available models (an industrial white paper)	Informal survey (review)	6	9	Compared nine TPI models w.r.t. four criteria: • The purpose of the model is to improve the whole test process • The model must have a maturity structure • There is sufficient information available about the model to compare it usefully • Model is current and is being updated since initial development
	[39]	2014	Adopting the right software test maturity assessment model (an industrial white paper by Cognizant Co.)	Informal survey (review)	0	2	The study concluded that: "*There is no single-fit right model*" and that enterprises should consider the following key questions when adopting the 'right' software TMA model: • What are the business drivers and objectives? • What is the kind of application being tested? • Is IT consolidated or distributed? • What is the purpose and the expected outcome of the test maturity assessment? • What is the relevance of the organization's IT products in the market, versus its competition's? • What is the long-term plan for the testing services?
	[13]	2014	Test process models: systematic literature review	SLR	48	23	Also extracted the source models used in the development of the new models, and the domains of the models (e.g., general, medical, embedded system)
	[12]	2016	Software test process improvement approaches: a systematic literature review and an industrial case study	SLR	31	18	The study found that "*many of the STPI approaches do not provide sufficient information or the approaches do not include assessment instruments*" [12]. This makes it difficult to apply many approaches in industry. Greater similarities were found between TPI Next and TMMi and fewer differences. The study concluded that numerous TPI approaches are available but not all are generally applicable for industry. A case study was conducted, afterwards, to identify TPI approaches valuable for a case organization, apply them and compare their content and assessment results.
	This MLR	2016	Software test maturity and test process improvement: a multivocal literature review	MLR	181	58	Novelty of this work compared to the existing body of literature: • Including the grey literature • Extracting the base models (e.g., CMMI) used for developing the new models • Synthesizing drivers (needs), challenges and benefits of TMA/TPI • Synthesizing methods for TPI • Synthesizing types of development process models • Extracting attributes of case-study companies / projects under study

(*continued on next page*)

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

Table 1 (*continued*)

	Reference	Year	Title	Research methodology	# of primary studies / references	# of models reviewed	Other contributions
Review is only a part	[40]	2008	Evaluation approaches in software testing (only Section 2.3 "*Test Process Evaluation & Improvement*")	Informal survey (review)	74	5	Just a general/brief review and comparison of five models: TMM, TMMi, TPI, Metrics-based Verification & Validation Maturity Model (MB−V^2M^2) and Test Process Assessment Model (TPAM)
	[41]	2009	An evaluation framework for software test processes (the first part of this dissertation surveys the existing approaches)	Informal survey (review)	62	2	Just a brief review of two models (TMMi and TPI)
	[42]	2010	Systematic literature review of software process capability/ maturity models	Informal survey (review)	61	2	Was not focused on TMA/TPI models, but rather then entire spectrum of maturity models in SE. The results showed that "*there exist a large variety of models with a trend to the specialization of those models for specific domains*" and that "*most of those models are concentrated around the CMM/CMMI framework and the standard ISO/IEC 15,504 (SPICE)*".
	[43]	2013	Empirical analysis of the test maturity model integration (TMMi) (Table 1 provides a mini-SM)	Informal survey (review)	17	10	Classified ten models based on five criteria: (1) number of maturity levels, (2) terminology, (3) base models/standards, (iv) assessment method (mechanism), and (5) availability. The study then chose TMMi as the model to be applied in a single-object case study in a company.
	[44]	2013	Managing corrective actions to closure in open source software test process	Informal survey (review)	28	4	Classified four models (TMM, TMMi, TPI and TIM) based on several criteria, e.g.: (1) number of levels, (2) number of key process areas, (3) assessment type (questionnaire or checklist), and (4) assessment foundation (CMM, SPICE, practical experience)

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

Table 1 a summary of other contributions made in each source (see column '*Other contributions*').

As we can see in Table 1, in terms of three aspects (the number of reviewed models, the number of sources reviewed, and also the scale/types of other contributions), and as we will see in the rest of this paper, our MLR is in fact the most comprehensive review study in this domain up to now. For example, we systematically synthesize and present the drivers (needs) for TMA/TPI, challenges, benefits, and methods for TPI, while none of these aspects has been covered by any of the previous review studies. In fact, the need for this MLR arose in the course of several ongoing industry-academia collaborative projects in which the authors have been involved and we wanted to rigorously assess and characterize the drivers (needs) for TMA/TPI, and its challenges and benefits. However we could not find any systematic synthesis of those in the literature. Another major advantage of our study is that we include the grey literature (practitioners' voice and contributions) in this work. All the above reasons have raised the need for another review paper on this topic.

3. Research method

In the following, an overview of our research method and then the goal and review questions of our study are presented.

3.1. Overview

As discussed in Section 2.2, MLRs have only recently started to appear in SE, i.e., since 2013 in [7]. Thus, no specific guidelines for conducting MLRs in SE have been proposed yet. However, given the similarity between SLR, SM and MLR studies, we benefitted from SLR guidelines proposed by Kitchenham and Charters [31] and SM guidelines by Petersen et al. [32, 33]. There exists no guideline for conducting MLRs in SE yet. In other fields, there exist papers which provide suggestions on how to conduct MLRs, e.g., in medicine [27] and education sciences [8], which we benefitted from.

We should note that certain phases of MLRs are quite different than regular SLRs, e.g., searching for and synthesizing grey literature. In those cases, we had to refer to the MLR heuristics in other fields, e.g., [8, 27], develop our own heuristics for those specific phases as we dealt with them in the study, and/or refer to the first author's recent experience in a recent MLR [10]. Last but not the least, for the aspects common to SLR, SM and MLR studies, e.g., analysis of the sources from the formal literature, we benefitted from our past experience in SM and SLR studies, e.g., [45-49]. In this outset, we should mention that, based on the need that there exists no specific guidelines for conducting MLRs in SE, the first two authors and a colleague of theirs are currently developing such a guideline which will be presented to the community in near future.

Using the above experience and guidelines, we developed our MLR process, as shown in Fig. 4. We discuss the MLR planning and design phase (its goal and RQs) in the next section. Sections 4 to 6 then present the follow-up phases of the process: source selection, development of the classification scheme (map) and then the systematic mapping, synthesis and review. As we can see, this process has a lot of similarity to the typical SLR processes [31] and also SM processes [32, 33], the major different being only in the handling of the grey literature (i.e., searching for those sources, applying inclusion/exclusion criteria on and synthesizing them). Also, similar to the way many of the previous SLRs in the SE have been conducted, the classification and data extraction phases of our MLR are essentially conducted via a SM which enables us answer a subset of the study's RQs. Then we conducted data synthesis (details to be discussed in Section 5.2) to answer the other RQs which required synthesis of data.

3.2. Goal and review questions

The goal of this study is to systematically map (classify), review and synthesize the state-of-the-art and –practice in the area of software test maturity assessment and test process improvement, to find out the recent trends and directions in this field, and to identify opportunities for future research, from the point of view of researchers and practitioners. Based on the above goal, we raise 12 review questions (RQs) grouped under three categories:

3.2.1. Group 1-Common to SM studies

The two RQs under this groups are common to SM studies and have been studied in previous work, e.g., [45–49]. Let us note that replication is not applicable and thus not intended in this context. The two following RQs are "classification" types and are raised to enable the classification of studies in this area. There is no classification data of this type for the area of TMA/TPI in previous studies to compare our data to.

- **RQ 1.1: Mapping of studies by contribution facet:** What are the different contributions by different sources and what aspects of TMA/TPI have been investigated so far? How many sources present test maturity/ process improvement models, methods, methodologies, tools, metrics, processes also, informal heuristics or guidelines?
- **RQ 1.2: Mapping of studies by research facet:** What type of research methods have been used in the studies in this area? Some of the studies present solution proposals or weak empirical studies where others presented strong empirical studies.

3.2.2. Group 2-Specific to the domain (TMA/TPI)

- **RQ 2.1-Proposed test maturity models:** What test maturity models have been proposed in the academic and grey literature?
- **RQ 2.2- Base maturity models used for developing new models:** Which maturity/ improvement models have been used or extended in the studies? We aimed to find and group the source models from RQ 2.1 that the newer test maturity and improvement models are based on.
- **RQ 2.3-Drivers:** What are the drivers (needs) for TMA/TPI?
- **RQ 2.4-Challenges:** What are the impediments (challenges) for TMA/TPI?
- **RQ 2.5-Benefits:** What are the benefits of TMA/TPI?
- **RQ 2.6-Methods for TPI:** What are the methods used for TPI?
- **RQ 2.7-Development process models:** What are the process models adopted in TMA/TPI studies? Some sources conduct TMA/TPI in the context of plan-driven (Waterfall) while others do so in the context of agile models.
- **RQ 2.8-Attributes of case-study companies / projects under study:** How many and what types of software systems or projects under analysis have been evaluated in each source? One would expect that a given paper or article applies the proposed idea to at least one system or project to show its effectiveness. The companies have been investigated in terms of number of projects, size, quantitative improvement level, application domain, and whether a hypothetical example or a real commercial case was studied.

3.2.3. Group 3- Attention to this topic from the research versus practitioner community

- **RQ 3.1-Attention level:** How much attention has this topic received from the research vs. practitioner community?

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

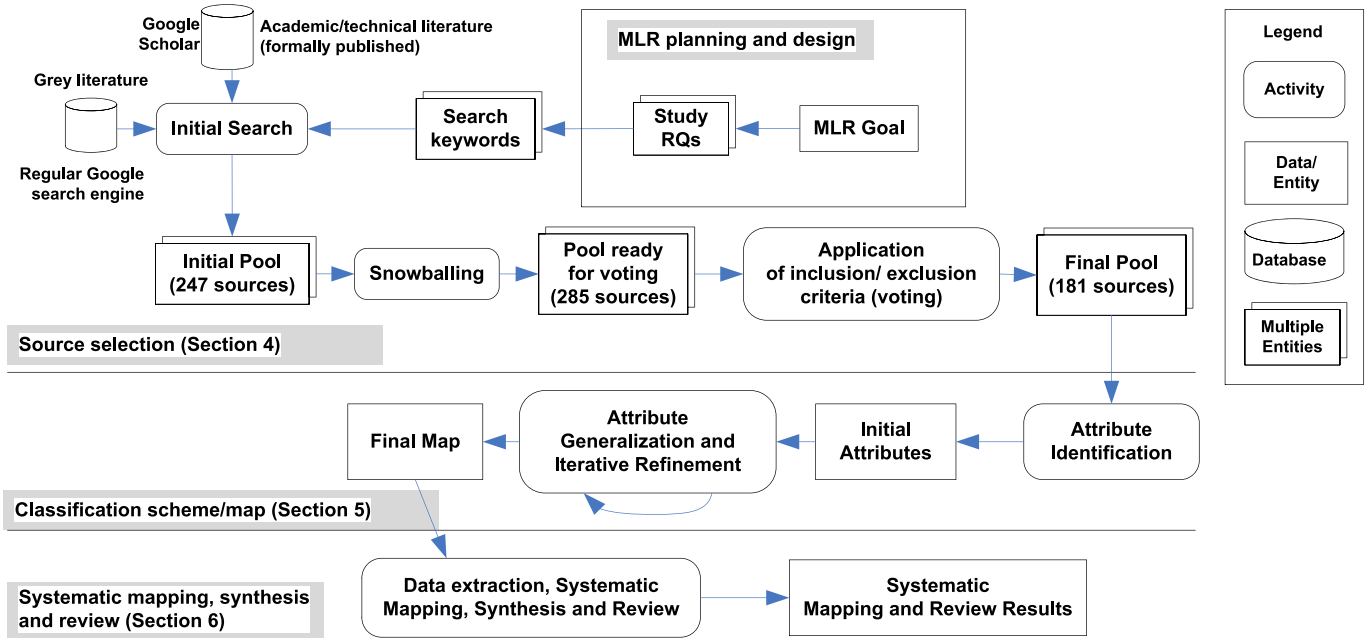

Fig. 4. An overview of our MLR process (as a UML activity diagram).

4. Searching for and selection of sources

Let us recall from our MLR process (Fig. 4) that the first phase of our study is article selection. For this phase, we followed the following steps in order:

- Source selection, search keywords and search approach (Section 4.1)
- Application of inclusion and exclusion criteria (Section 4.2)
- Finalizing the pool of articles and the online repository (Section 4.3)

4.1. Source selection, search keywords and search approach

We used the Google Scholar to search for the scientific literature, and use the Google's regular search engine for the grey literature. Our search strings in both search engines were:

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

All authors searched independently with the search strings, and in connection with the search the authors already applied inclusion/exclusion criterion for including only those which explicitly addressed either of the studies topics (test maturity or test process improvement).

When searching for the grey literature in Google's regular search engine, we utilized the relevance ranking of the search engine (Google's PageRank algorithm) to restrict the search space. This heuristic has also been utilized in previous MLRs and MLR-related experience papers in other fields, e.g., [50, 51]. For example, if one applies the above search string (a) (software test maturity) to the Google search, 1710,000 results would show as of this writing (Dec. 2015), but as per our observations, relevant results usually only appear in the first few pages. Thus, we checked the first 10 pages (i.e., somewhat a search "saturation" effect) and only continued further if needed, e.g., when the results in the 10th page still looked relevant. Note that all the decisions (albeit small) for the entire process were made with a majority voting among the researchers and no single researcher decided the actions on her/his own.

To ensure including all the relevant sources as much as possible, we conducted forward and backward *snowballing* [52], as recommended by systematic review guidelines, on the set of papers already in the pool. Snowballing, in this context, refers to using the reference list of a paper (backward snowballing) or the citations to the paper to identify additional papers (forward) [52]. Snowballing (both backward and forward) was done after populating the final list of primary studies and after applying inclusion/exclusion and quality assessment criteria. To keep our efforts manageable, snowballing was conducted on a random set of 10 papers in the initial pool. We used a random number generator to pick those sources from the pool.

As the MLR process (Fig. 4) showed, our initial search in the search engines yielded 247 sources. Snowballing [52] added 38 sources, bringing the total initial pool count up to 285 sources, e.g., [Source 24] and [Source 111] were found by backward snowballing of [Source 43], i.e., the latter source had cited the former sources.

4.2. Inclusion/exclusion criteria and voting

We carefully defined a set inclusion and exclusion criteria to ensure including all the relevant sources and not including the out-of-scope sources. The criteria were as follows:

- Criterion #1: Is the source relevant to our scope (test maturity assessment or test process improvement)?
- Criterion #2: Does the paper include a relatively sound validation (for grey sources: does the source provide interesting addition to our pool)?
- Criterion #3: Is the source in English and can its full-text be accessed?

The last author reviewed each candidate source and placed her opinion w.r.t. the criteria #1 and #2 in an online spreadsheet. Then the other two authors reviewed the votes of the third author and we ensured to reach a consensus on each and every vote. 'Relevancy' of each source to our scope was decided whether the given source (study) had a focus on TMA/TPI and contributed any material (e.g., model, tool, technique or empirical findings) in this area.

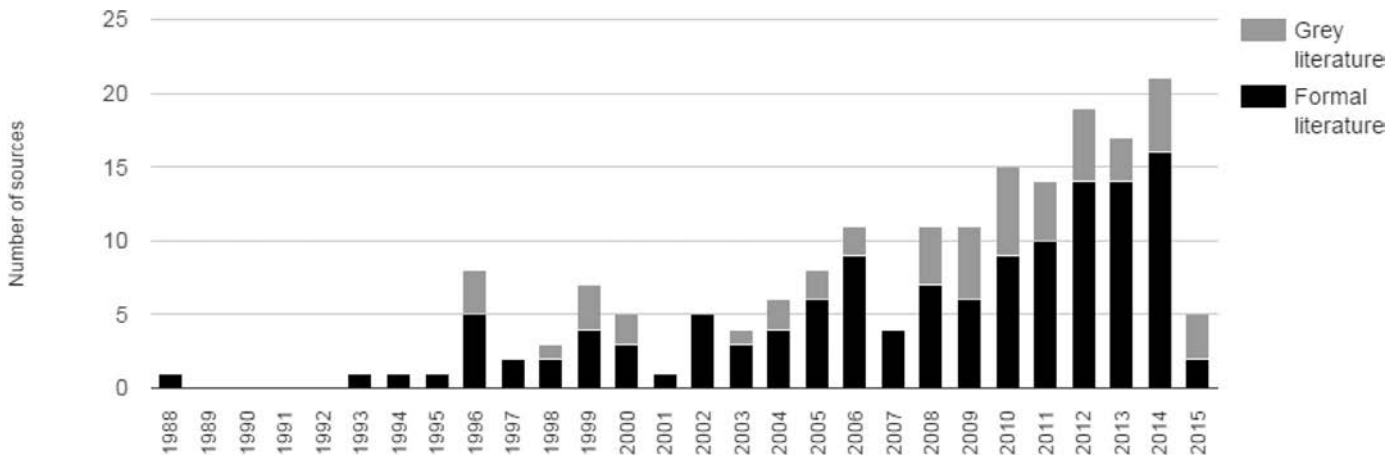

Fig. 5. Growth of the TMA/TPI field and types of the sources (formally published vs. grey literature) across different years.

The criterion #2 above is actually a quality assessment criterion used frequently in SM and SLR studies and, at the same time, also served as an inclusion/exclusion criterion in our study. The answer for each question could be {0, 1}. Each of the researchers voted for each of the criteria independently. After the voting phase ended, the researchers discussed differences in the votes to reach unanimous decisions for all the sources. Only the sources which received 1′s for all three criteria were included. The rest were excluded.

4.3. Final pool of sources and the online repository

From the initial pool of 285 sources which all authors voted on, 104 sources were excluded during voting. This finalized the pool with 181 sources, from which 130 (71%) were formally published sources (e.g., conference and journal papers) and 51 (29%) were sources in the grey literature (e.g., internet articles and white papers). The final pool of sources, the online mapping repository and details on why each individual source was excluded can be found in the study's online Google spreadsheet [53].

In Fig. 5, we show (as a stack chart) the plot of annual number of sources by type (formally published vs. grey literature). As we can see, the attention level in this topic has steadily risen since early 1990′s by both the research and practitioner communities, as shown by the number of sources published each year. Note that, for the year 2015, our pool contains only 5 sources (is partial), since the source selection of the study was conducted in June of 2015.

5. Development of the systematic map and data-extraction plan

To answer each of the MLR's RQs, we conducted a SM first, in which we developed a systematic map and then extracted data from papers to classify them using it. We then conducted qualitative synthesis to answer the RQs. Details are discussed next.

5.1. Development of the systematic map

To develop our systematic map, we analyzed the studies in the pool and identified the initial list of attributes. We then used attribute generalization and iterative refinement to derive the final map.

As studies were identified as relevant to our study, we recorded them in a shared spreadsheet (hosted in the online Google Docs spreadsheet [53]) to facilitate further analysis. Our next goal was to categorize the studies in order to begin building a complete picture of the research area and to answer the study RQs. We refined these broad interests into a systematic map using an iterative approach.

Table 2 shows the final classification scheme that we developed after applying the process described above. In the table, column 2 is the list of RQs, column 3 is the corresponding attribute/aspect. Column 4 is the set of all possible values for the attribute. Column 5 indicates for an attribute whether multiple selections can be applied. For example, in RQ 1.1 (research type), the corresponding value in the last column is 'S' (Single). It indicates that one source can be classified under only one research type. In contrast, for RQ 1.2 (contribution type), the corresponding value in the last column is 'M' (Multiple). It indicates that one study can contribute more than one type of options (e.g. method, tool, etc.). Finally, the last column denotes the answering approach for each RQ, whether a SM (classification) was enough or qualitative coding (synthesis) had to be conducted, as discussed next.

5.2. Data extraction and synthesis

Once the systematic map was developed, each of the researchers extracted and analyzed data from the subset of the sources assigned to her/him. Before each researcher extracts all the sources in her/pool, a pilot phase was first conducted on a set of five papers to come to a general consensus on how each paper needed to be classified, and to ensure that the authors would share a common understanding of terminology and classifications. We included traceability links on the extracted data to the exact phrases in the sources to ensure that how the classification is made is suitably justified. For each source, we assigned a second person as a peer reviewer to check and, in case of disagreements, discussions were made between the two or even three authors, to correct the extracted data in coordination with the person who had initially extracted the data.

Fig. 6 shows a snapshot of our online spreadsheet hosted on Google Docs that was used to enable collaborative work and classification of sources with traceability links (as comments). In this snapshot, classification of sources w.r.t. RQ 1.1 (Contribution type) is shown and one researcher has placed the exact phrase from the source as the traceability link to facilitate peer reviewing and also quality assurance of data extractions.

After all researchers finished data extractions, we conducted systematic peer reviewing in which researchers peer reviewed the results of each other's analyses and extractions. In the case of disagreements, discussions were conducted. This was conducted to ensure quality and validity of our results. Fig. 7 shows a snapshot of how the systematic peer reviewing was done. In total, out of the 181 sources, the joint peer reviewing efforts identified corrections

Table 2
Systematic map developed and used in our study.

Group	RQ	Attribute/Aspect	Categories	(M)ultiple/ (S)ingle	Answering approach
Group 1-Common to all SM studies	1.1	Contribution type	Heuristics/guideline, method (technique), tool, metric, model, process, empirical results only, other	M	SM
	1.2	Research type	Solution proposal (simple examples only), validation research (weak empirical study), evaluation research (strong empirical study), experience studies, philosophical studies, opinion studies, other	S	SM
Group 2-Specific to the domain (TMA/TPI)	2.1	Proposed test maturity models	Name of test maturity models proposed in the papers	M	SM
	2.2	Source maturity models used	{TMMi, TMM, CMMI, CMM, TPI, TPI-Next, TestSPICE, TMap, TIM, Other}	M	SM
	2.3	Drivers	Textual explanation of the TMA/TPI drivers as mentioned in papers	M	Qualitative coding
	2.4	Challenges	Textual explanation of the TMA/TPI challenges as mentioned in papers	M	Qualitative coding
	2.5	Benefits	Textual explanation of the TMA/TPI benefits as mentioned in papers	M	Qualitative coding
	2.6	Methods for TPI	{Using a maturity/TPI model, Using metrics without a model, control theory, simulation, other}	M	SM
	2.7	Development process models	{Plan-driven, Agile / iterative, other, not explicit}	M	SM
	2.8	Attributes of case-study companies / projects under study	# of cases (companies, projects): integer Names of company / software project(s): String Size of company (# of employees): integer # of subjects (for surveys): integer Quantitative improvements, if any Any other metrics measured (if any) Application domain: String Type of software project: {Hypothetical / simple (toy) example, commercial / real}	M	Just the metrics
Group 3-Trends and demographics	3.1	Attention level	{A: Academic, I: Industry, C: collaboration}	S	Just the metric
	3.2	Citations to technical papers	Citation count form Google Scholar on Dec. 20, 2015	S	Just the metric

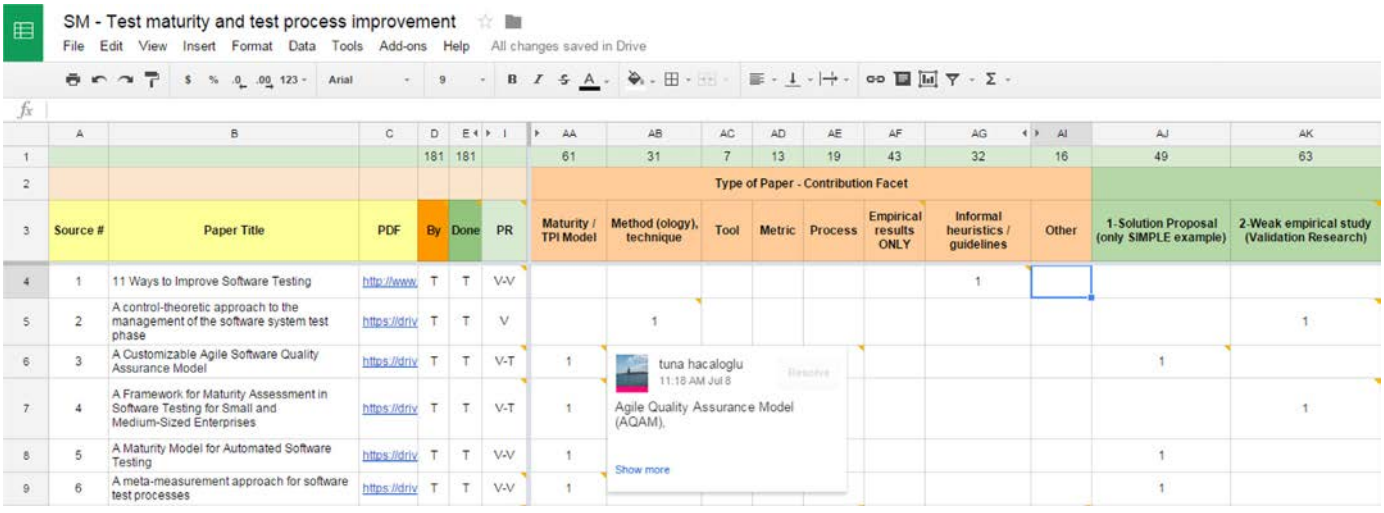

Fig. 6. A snapshot of the publicly-available spreadsheet hosted on Google Docs that was used to enable collaborative work and classification of sources with traceability.

to be made on data extracted for 54 sources (29.8% of the pool). Details can be found in the study's online Google spreadsheet [53].

As shown in Table 2, to address three of the study's RQs, RQ 2.3 (Drivers), RQ 2.4 (Challenges) and RQ 2.5 (Benefits), we had to conduct qualitative coding (synthesis) of data. To choose our method of synthesis, we carefully reviewed the research synthesis guidelines in SE, e.g., [54–56], and also other SLRs which had conducted synthesis of results, e.g., [57, 58]. According to [54], the key objective of research synthesis is to evaluate the included studies for heterogeneity and select appropriate methods for integrating or providing interpretive explanations about them [59]. If the primary studies are similar enough with respect to interventions and quan-

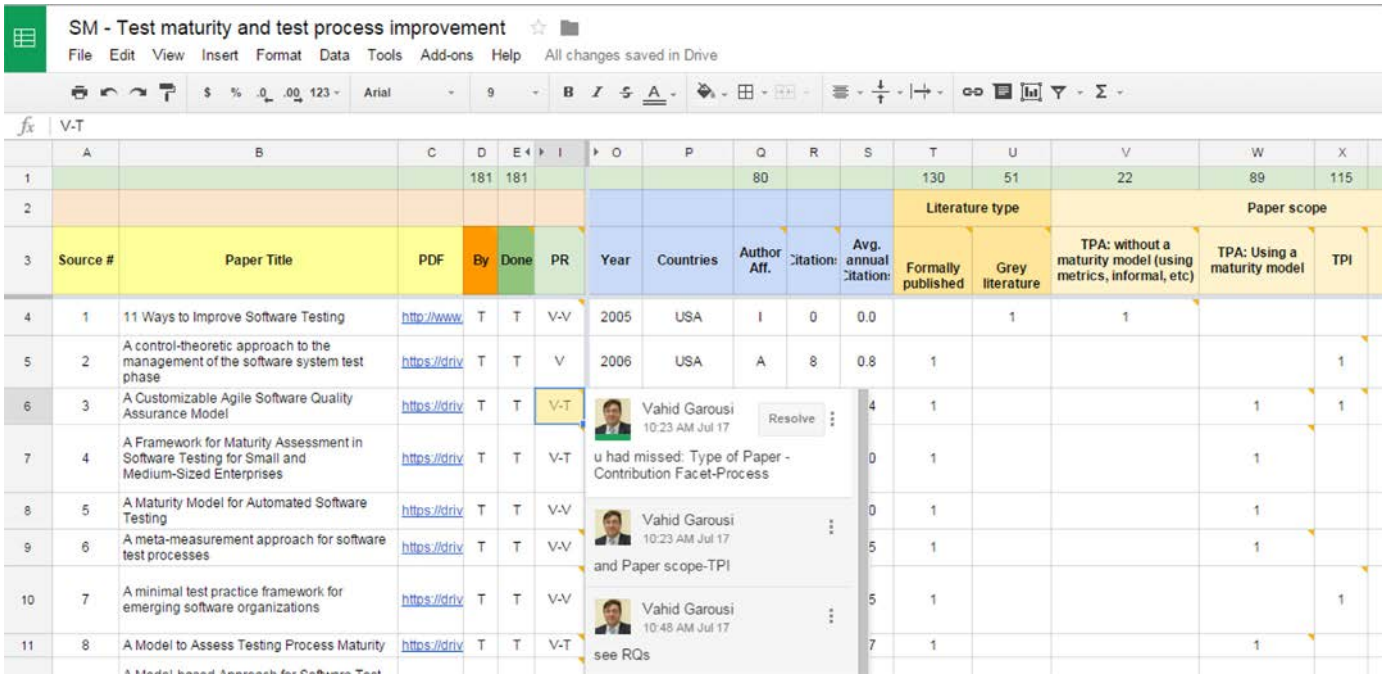

Fig. 7. A snapshot showing how the systematic peer reviewing was done.

titative outcome variables, it may be possible to synthesize them by meta-analysis, which uses statistical methods to combine effect sizes. However, in SE in general and in our focused domain in particular, primary studies are often too heterogeneous to permit a statistical summary. Especially for qualitative and mixed methods studies, different methods of research synthesis, e.g., thematic analysis and narrative synthesis, are required [54].

The classification presented in the rest of this paper for 'drivers', 'challenges' and 'benefits' were the results of a formal and systematic synthesis done collaboratively between the three researchers following a systematic qualitative data analysis approach [60]. We had some pre-defined initial set of classifications for these aspects (based on our past knowledge of the area), e.g., cost and time for drivers. During the qualitative data analysis process, we found out that our pre-determined list of factors had to be expanded, thus, the rest of the factors emerged from the sources, by conducting "open" and "axial coding" [60]. The creation of the new factors in the "coding" phase was an iterative and interactive process in which all three researchers participated. Basically, we first collected all the factors related to questions RQs 2.3, 2.4 and 2.5 from the sources. Then we aimed at finding factors that would accurately represent all the extracted items but at the same time not be too detailed so that it would still provide a useful overview, i.e., we chose the most suitable level of "abstraction" as recommended by qualitative data analysis guidelines [60]. Fig. 8 shows a snapshot of qualitative coding of data to answer those RQs.

6. Results

Results of the systematic mapping are presented in this section from Sections 6.1 to 6.3.

6.1. Group 1-RQs common to all SM studies

6.1.1. Mapping of sources by contribution facet (RQ 1.1)

Fig. 9 shows the growth of the field based on the mapping of sources by contribution facet. Note that the numbers on the Y-axis are cumulative. 58 sources contributed TMA/TPI models. 38 sources contributed methods, methodologies, techniques, or approaches (we considered all of these in the same category). 6 sources contributed tool support. 13 and 18 sources contributed metrics and processes, respectively. 44 contributed empirical results mainly. 28 and 20 contributed informal heuristics / guidelines, and 'Other' types of contributions, respectively. We discuss next excerpts and example contributions from each category of contributions.

6.1.1.1. TMA/TPI models. The list of TMA/TPI models proposed or used (for empirical studies) in the sources will be comprehensively discussed in Section 6.2.

6.1.1.2. Methods, methodologies, approaches, techniques. 38 sources contributed methods, methodologies, techniques, or approaches. For example, [Source 6] proposed a meta-measurement approach / framework for TMA/TPI, consisting of the following components: target (software test processes), evaluation criteria (quality attributes), reference standard (process measurement profiles), assessment techniques (test process measurements), evaluation process (guidelines and procedures for the whole evaluation process). [Source 19] contributed an approach called "Turbo-team" for TPI which a quick, quantified approach to working in teams that emphasizes empowering the right mix of people to focus rapidly on a well-bounded problem/opportunity, while leveraging off of past team successes and lessons learned.

[Source 27] contributed a simulation approach for hardware/software co-design and the field testing of critical embedded software systems. [Source 59] proposed a strategy for improving scenario-based testing processes by offering a value-based testing prioritization strategy which allows tests to be ranked by how well the tests can reduce risk exposure. In [Source 62], the authors extended and elaborated the '4 + 1' theoretical view of Value-based Software Engineering (VBSE) framework in the software testing process and propose a multi-objective feature prioritization strategy for testing planning and controlling, which aligns the internal testing process with value objectives coming from customers and markets.

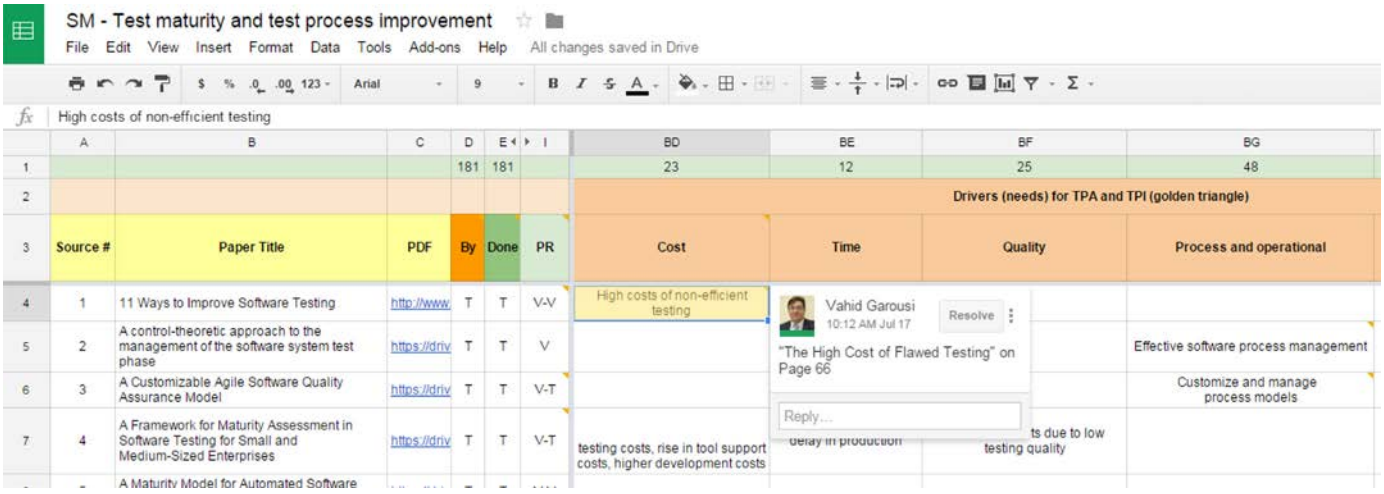

Fig. 8. A snapshot showing qualitative coding of data to answer RQs.

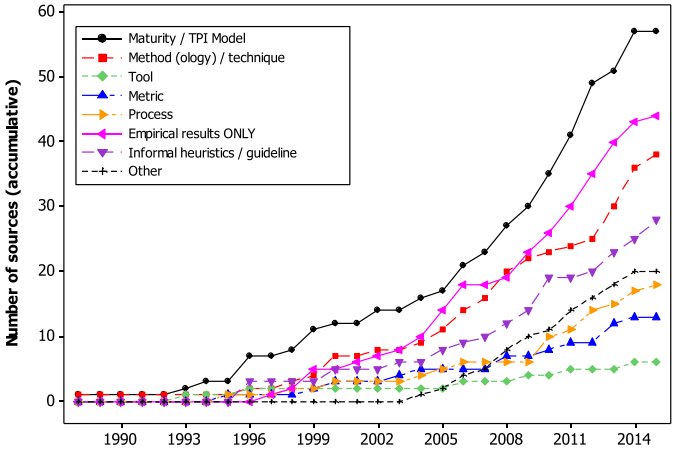

Fig. 9. Growth of the field based on the mapping of sources by contribution facet.

[Source 72] proposed an OSS test process assessment framework (abbreviated as OSS-TPA) that provides guidelines, procedures, and metrics with the aim of evaluating testing in open-source software (OSS) projects.

6.1.1.3. Tools. Six sources contributed tool support. In [Sources 44 and 46], the authors argue that testing process maturity growth should be supported by automated tools. Then developed a tool-set called "Testers' Workbench" to provide that support. [Source 80] contributed a test project assessment application. The application helps in evaluating projects under various companies using TMMi levels and standards and hence, generating reports in form of graphs showing the areas that need to have improvement. [Source 86] contributed an automated method and system for evaluating an organization's testing maturity capabilities.

6.1.1.4. Metrics. 13 sources contributed metrics. For example, [Source 6] contributed a metric called private quality index (P.QI). [Source 12] proposed a set of recommended measurements for each maturity goal of TMM based on the Goal/Question/Metrics (GQM) framework. Several other new metrics were proposed in [Source 37], e.g., test improvement in product quality, test time needed normalized to size of product, test cost normalized to size of product, and cost per weighted defect unit. [Source 59] contributed another metric called Average Percentage of Business Importance Earned (APBIE).

6.1.1.5. Processes. 18 sources contributed processes. For example, [Source 3] a contributed process for a customizable Agile Software Quality Assurance Model (AQAM). [Source 7] proposed a process for introduction of new test methods in a company. [Source 58] argued that the process of TPI is similar to any other improvement process and depicted it as we showed in Fig. 3 in Section 2.1 of our paper. [Source 75] presented a general process of test enhancement methods.

6.1.1.6. Empirical results only. Primary contributions of 44 sources were empirical results. For example, [Source 9] empirically explored and described how risk is defined, assessed and applied to support and improve testing activities for industrial projects, products, and processes. [Source 15] reported the results of a survey on software test maturity in Korean defense industry. [Source 30] reported the results of a survey on barriers to implement test process in small-sized companies.

To characterize the state of the practice in software testing through a TMMi-based processes, [Source 31] reported the results of a survey amongst Brazilian software testing professionals who work in both academia and industry, in order to identify priority practices to build the intended, streamlined processes. [Source 22] was a Masters' thesis which examined and evaluated the maturity level of testing processes of a given company (HR System Solutions) are at, and what necessary actions should be focused on in order to reach a more mature level of testing.

[Source 23] investigated a specific test process called CenPRA (Centro de Pesquisas Renato Archer), a generic software testing model defined by selecting software testing "best practices" ; with the following aspects: (1) an evaluation of the CenPRA test process under the perspective of CMMI, (2) an evaluation of which aspects of CMMI are taken into account by the CenPRA test process, and (3) an evaluation of how the CenPRA model can be used to supplement software testing related aspects of CMMI.

[Source 37] presented the results of applying a single quality metric versus a set of complementary metrics to measure the test process of the IBM Electronic Commerce Development (ECD) test teams, and analyzed the effectiveness of a "metric set" to measure the test process. [Source 49] was a qualitative study of ten software organizations to understand how organizations develop their test processes and how they adopt new test methods.

6.1.1.7. Informal heuristics /guidelines. 28 sources contributed informal heuristics / guidelines. For example, [Source 1] contributed a list set of to-do's such as: (1) respect your testers, (2) Co-locate

your testers and developers, and (3) Apply equivalence class partitioning. [Source 31] was book entitled "Critical Testing Processes: Plan, Prepare, Perform, Perfect". The author mentioned that: *"The processes in this book aren't pie-in-the-sky theory, but rather grew out of my experiences on the ground as a practicing tester, test lead, and test manager. Your experiences and your challenges will differ from mine, so be sure to adapt my processes—or completely reinvent your own—rather than trying to put a saddle on a cow. How do we recognize good processes? We can look for test team behaviors and achievements that indicate the degree of success of the process".*

[Source 49] explained how to achieve the CMM levels for QA processes, explained with best examples. [Source 69] suggested that the effectiveness of the use and management of the organizational test specification should be considered and any feedback and changes to improve its effectiveness should be determined and approved. [Source 94] provided reflections on TPI and careers of test managers. [Source 95] provided informal recommendations for application of TMMi as follows: "*We have also found that although the model is quite easy to understand, implementing it in an organization and thereby improving the test process is not always a straightforward task. On averages it take approximately two years to achieve TMMi level 2*".

[Source 109] was a case study of TMM at Marks & Spencer Corp. in the UK. The authors suggested that conducting TMM assessment is "*time consuming, but comprehensive*". They suggested that: "*Ensure that you have buy-in from everyone*" and highlighted the importance of "*Communication, communication, communication*".

6.1.1.8. Other. 20 sources contributed 'Other' types of contributions. For example, [Source 14] contributed an ontology for a TMA model called *MND-TMM* (Test maturity model for the Korean ministry of Defense). [Source 39] discussed the need for development of TMM and correlation between CMM and TMM. [Source 60] explored the applicability of various models. [Source 61] presented a cause-effect graph of TPI factors. [Source 64] analyzed the mapping / correlation between TMMi and TPI-Next. [Source 75] proposed a "recommendation system" for TPI. The recommendation system suitably recommends enhancements according to the classification of the tests. The tests may be modified accordingly.

[Source 76] contributed an assessment guideline, organized into four phases: (1) hire assessor, (2) prepare assessment, (3) conduct assessment, and (4) document assessment. [Source 78] a contributed checklist for testing open-source software (OSS) systems. [Source 89] discussed impediments (challenges) for TPI and proposed/applied solutions. [Source 90] contributed an improvement (transformation) TPI roadmap for a specific industrial client in the area of financial services.

6.1.2. Mapping of sources by research facet (RQ 1.2)

Fig. 10 shows the mapping of sources by research facet. Note that this classification was done similar to our past experience in earlier SM and SLR studies, e.g., [45–49]. Categories 1, 2 and 3 correspond to solution proposal (simple examples only), weak empirical study (validation research), and strong empirical study (evaluation research), i.e., in increasing levels of research approach maturity. Solution proposal sources only provide simple examples while weak empirical studies are shallower in comparison to #3 with no hypothesis testing and no discussions of threats to validity. However, strong empirical studies include RQs and hypothesis testing and also discussions of threats to validity.

As we can see in Fig. 10, a large number of the studies (63 of them) were classified as weak empirical studies which is a quite good indication of research rigor in this area of software testing, even given the fact that 51 sources in our pool (29%) were form the grey literature, and which generally do not use rigorous research approaches.

Experience studies were those who had explicitly used the phrase "experience" in their title or in discussions frequently without conducting an empirical study. There were 30 such sources. For example, [Source 89] proposed ten factors that impede improvement of verification and validation processes in software organizations. The paper said that: "*These factors are obtained from the authors' experience in several software process improvement initiatives in software verification and validation processes*" .

We classified 4 and 14 sources, respectively, under philosophical and opinion studies. For example, [Source 94] was a reflection on TPI and career of test managers and we considered it philosophical. [Source 93] was titled *'Test process Improvement and Agile Friends or foes?*' and we considered it an opinion since it had the following statement inside: "*Coming more from a practical background and approaching this with an open mind, I strongly beg to differ. Many organizations struggle when they are in transition from a sequential life cycle to an Agile iterative life cycle*".

To put the research facet breakdown of this area in context, we compare the ratios with research facet classifications reported for four other SE areas as reported in four previous SM studies: Software Test-Code Engineering (STCE) [46], web application testing [61], Graphical User Interface (GUI) testing [62], and Dynamic Software Product Lines (DSPL) [63]. Fig. 11 shows the comparisons. In terms of research rigor (solution proposal, weak and strong empirical research), the STCE field seems to be first rank. Then, the other areas position quite similarly. The current area (TMA/TPI) has a high ratio of 'experience' -based sources mainly due to including the grey literature.

6.2. Group 2- RQs specific to the domain (TMA/TPI)

6.2.1. Proposed test maturity models (RQ 2.1)

Our first 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 and when it used an existing model. 58 of the 181 sources presented new TMA/TPI models. 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 a few examples in Table 3, while the full list can be found in the online spreadsheet [53]. We also mention the levels of the 'staged' TMA/TPI models in Table 3.

In terms of popularity, TMMi (and its earlier version TMM) [Source 91] and TPI (and its successor TPI-Next) [Source 58] are the most popular models. TMMi and TMM have been used for assessments or as base models in 58 sources while TPI and TPI-Next have been used for those purposes in 18 sources. 28 sources used other models for TMA/TPI, e.g., TestSPICE [Sources 71, 88, 99, 101], TMap [Source 108].

We are observing the development of models such as TPI-EI [Source 20] which is the adoption of the TPI model in the embedded software domain, the Unit Test Maturity Model [Source 107], or the Personal Test Maturity Matrix [Source 104] which is used to gauge test engineers' 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') [64], the testing maturity models are also, broadly speaking, fall under either of these two types. For example, TMMi, AQAM and ASTMM 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, 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.

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

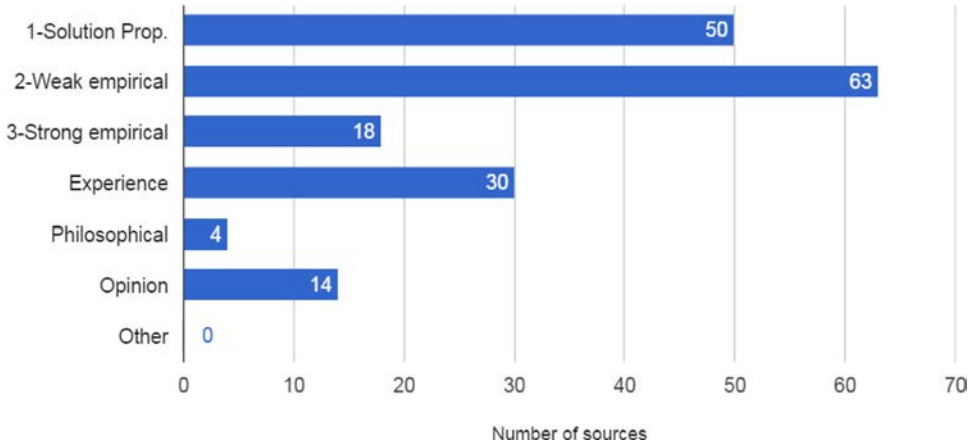

Fig. 10. Mapping of sources by research facet.

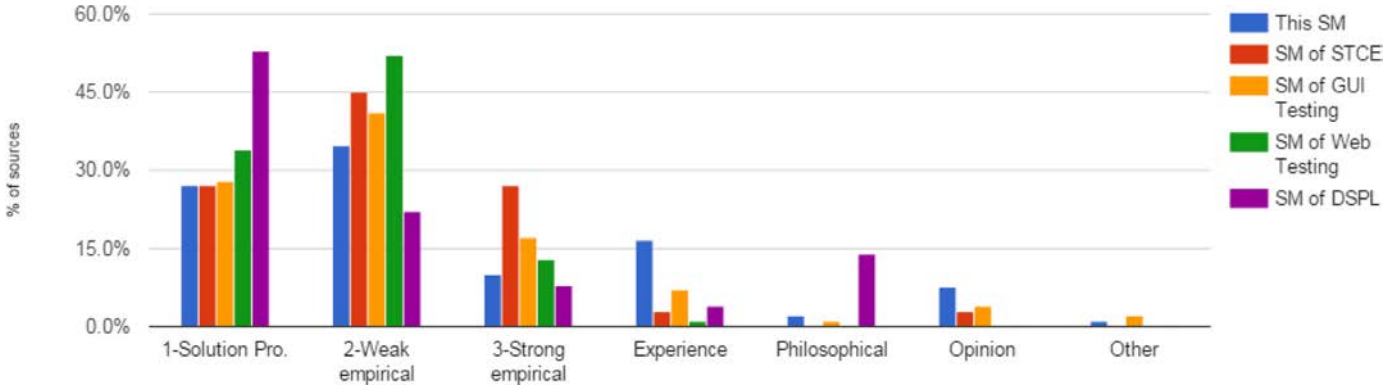

Fig. 11. Comparisons of the composition of research facets in the TMA/TPI field versus four other SE areas.

Table 3
Examples of the test maturity models proposed in the community along with their maturity levels.

Test Maturity Model integration (TMMi) [Source 91] • Level 1: Initial • Level 2: Definition • Level 3: Integration • Level 4: Management and measurement • Level 5: Optimization	TPI (Test process improvement) [Source 58]: 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 …. 18. Test process management 19.Evaluation 20. Low-level testing	Unit Test Maturity Model [Source 107] • 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
Agile Quality Assurance Model (AQAM) [Source 3] • Level 1: Initial • Level 2: Performed • Level 3: Managed • Level 4: Optimized	Automated Software Testing Maturity Model (ASTMM) [Source 5] • Level 1: Accidental automation • Level 2: Beginning automation • Level 3: Intentional automation • Level 4: Advanced automation	TPI-EI [Source 20] Adaptation of TPI for embedded software
Agile Testing Maturity Model (ATMM) [Source 29] • Level 0: Waterfall • Level 1: Forming • Level 2: Agile bonding • Level 3: Performing • Level 4: Scaling	TestSPICE [Source 71] A set of KPAs. Based on ISO/IEC 15504, Software Process Improvement and Capability dEtermination (SPICE) standard	The Personal Test Maturity Matrix [Source 104] A set of KPAs such as: test execution, automated test support and reviewing

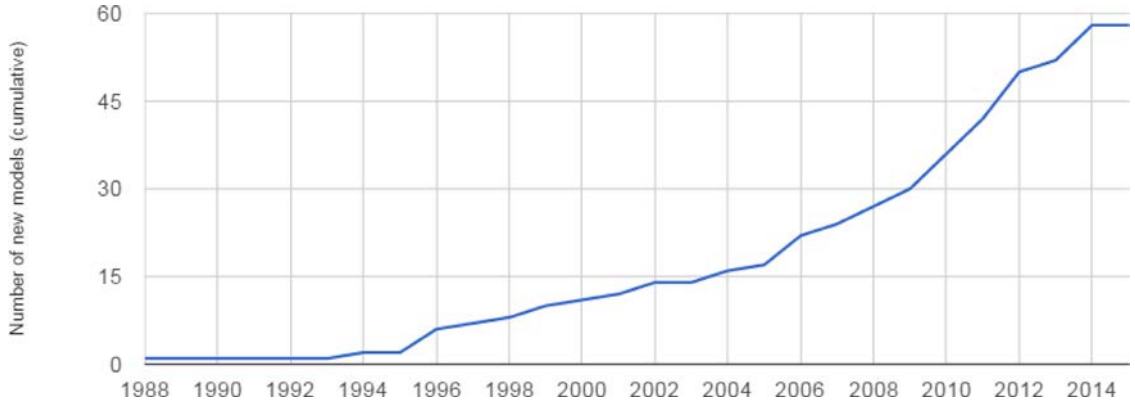

Fig. 12. Growth trend of the newly proposed TMA/TPI models in the sources over the years.

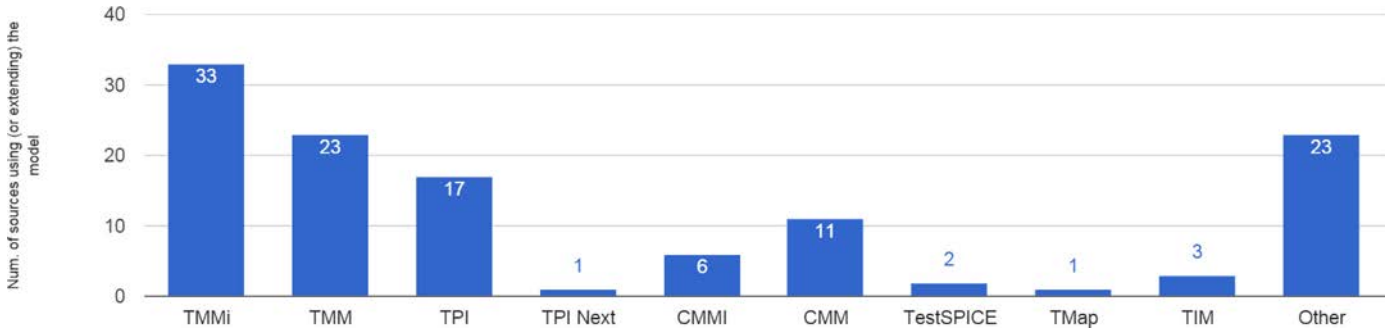

Fig. 13. Breakdown of the frequency of the source maturity model used in the sources.

What is 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 the 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 need, but rather on hypothetically argued motivations and also often by not fully reviewing what is already out there to ensure minimizing the overlap.

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 [43] and also experienced by the authors in their ongoing industrial projects, e.g., [65] or [66]. To further add to the complexity of conducting TMA/TPI using these models, many have reported challenges when using even established models such as the TMMi [43], e.g., not being able to objectively assess each maturity area/item using the existing model guidelines. What we as a community need in this regard are more practical guidelines on how to choose the most suitable models as well as guidelines on how to effectively and efficiently apply them in practice.

Fig. 12 depicts, as a cumulative chart, the growth trend of the proposed test maturity models in the sources over the years. As we can see, the trend shows that new models have been added throughout the years quite constantly.

6.2.2. Based maturity models used for developing new models (RQ 2.2)

117 sources used the existing models for TMA/TPI purposes to build newer (better or more specific) models. Fig. 13 shows the breakdown of the frequency of the source maturity model used in the sources. TMMi and its earlier version TMM were used the highest, in 34 and 23 sources, respectively. TPI was the 3rd in the rank.

For example, [Source 52] reported a survey to identify the key practices to support the definition of a generic streamlined software testing process, based on the practices described in the TMMi. [Source 67] used TMMi as follows. It argued that if an organization intends to evaluate and improve its test process based on TMMi, then identifying and controlling product risks—which can be ensured by integrating risk-based testing into the test process—are necessary to reach TMMi level 2 (“Managed”).

Examples of the ‘other’ models used were the followings: ISO/IEC 15,504 Information technology – Process assessment, also termed Software Process Improvement and Capability dEtermination (SPICE) in [Source 8], ISO/IEC 29,119 in [Sources 11, 38], a custom-made model named Levels of Information Systems Interoperability (LISI) in [Source 16], safety lifecycle aspects from IEC 61,508 in [Source 28], and Software Capability Maturity Model (SWCMM) in [Source 68].

Fig. 14 shows the evolution of TMA/TPI models and their relationships. The idea for this figure has been inspired by a similar evolution model prepared for the UML.[1] The data for this visualization come form results of RQ 2.1 and 2.2 (older models used as bases for developing new models). As we can see, new 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 [Source 36], proposed in 2001, is based on TMM [Source 33].

6.2.3. Drivers (RQ 2.3)

All the above RQs were conducted with a SM (mapping) approach. To extract and synthesize TMA/TPI ‘drivers’ from sources, as discussed in Section 5.2, we conducted qualitative coding and results are presented next.

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

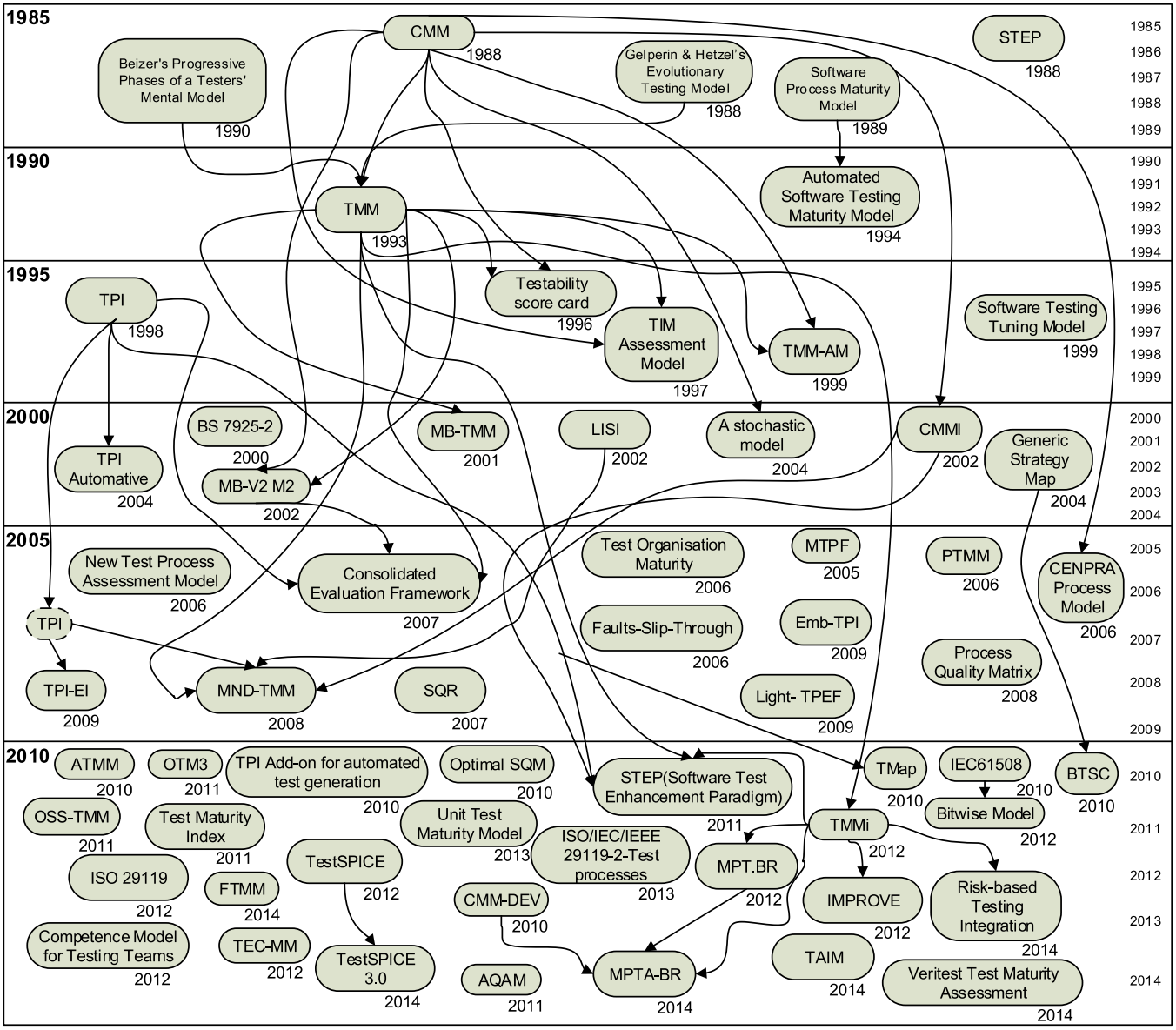

Fig. 14. Graph showing the evolution of TMA/TPI models and their relationships.

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 qualitative coding of the drivers as reported in the sources, we synthesized and classified drivers (needs) for TMA/TPI into five categories:

1. Process and operational needs (mentioned in 48 sources)
2. Needs related to software quality (25 sources)
3. Cost-related needs (23 sources)
4. Time and schedule-related needs (12 sources)
5. "Other" needs (18 sources)

We discuss examples from each category next.

6.2.3.1. Process and operational needs. 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 [Source 19], better test efficiency [Source 42], not meeting expectations or commitments [Source 44], internal stakeholder dissatisfaction [Source 44], missing exit criteria for testing [Source 54], lower risks, improve the productivity [Source 57], raise profile of testing, baseline test capabilities, and develop credible testing roadmap [Source 109].

For example, [Source 19] discussed the lack of a process improvement infrastructure at the Union Switch and Signal (US&S) and used that as a motivation to conduct TMA/TPI in that organization. As a paper entitled '*Experiences from informal test process assessments in Ireland: top 10 findings*', [Source 44] mentioned the following specific process and operational needs: not meeting expectations or commitments, the need to scale up operations and provide more structure, internal stakeholder dissatisfaction, poor management or operations visibility on project progress, and inconsistency across projects.

A paper entitled '*Identification of test process improvements by combining fault trigger classification and faults-slip-through measurement*' [Source 53] quote from a test manager at Ericsson: "*I would not only like to know which phases that need improvements; I also want to know which activities in each phase that should be improved*". In '*Improvement Areas for Agile Test Processes*' [Source 57], the drivers were identified as: making testing more efficient, low-

ering the risks, and improving the productivity. [Source 74] stated: "*there has always been a need to increase the efficiency of testing while, in parallel, making it more effective in terms of finding & removing defects*".

[Source 79] stated that: "*Processes and work agreements to be up-dated. Gaps in communication to be identified. Enhancement initiatives to be prioritized. The level of unit testing varies significantly. All repercussions of code changes are not known. No exit criteria defined for testing. Lack of proper test case specification. Test automation is not integrated with manual test process. Documentation processes are vague*". [Source 97] stated that: "*The primary driver for the test tuning was to identify the problems occurring in the field and direct the test effort appropriately*".

6.2.3.2. Needs related to software quality. Examples of needs related to software quality, as mentioned in the sources, are as follows: number of faults due to low testing quality [Source 4], direct relationship between the quality of the test process to the final quality of the developed product [Source 23], and lack of planning and resources for testing impacting software quality [Source 31].

6.2.3.3. Needs related to cost. Examples of cost-related needs are as follows: the argument that most current testing processes are often technique-centered, rather than organized to maximize business value [Source 62], testing costs being too high [Source 113], and low cost effectiveness of testing [Source 115].

Many sources reported 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 challenges and efforts to be spent on the activities and benefits relate to drivers and needs. Only if the expected benefits outweigh the costs, TMA/TPI activities will get the green light to start.

As a paper entitled '*Experiences from informal test process assessments in Ireland: top 10 findings*', [Source 44] mentioned the following cost-related needs: high priority business/department drivers such as cost reduction and productivity increases.

[Source 59] noted that "*Commercial experience is often that 80% of the business value is covered by 20% of the tests or defects, and that prioritizing by value produces significant payoffs*" .

[Source 61] mentioned that "*The costs of testing of a software project or product are considerable and therefore it is important to identify process improvement propositions for testing*" . Also, many believe that most current testing processes are often technique-centered, rather than organized to maximize business value [Source 62].

6.2.3.4. Needs related to time pressure and project schedule. Examples of schedule-related needs are as follows: delay in production due to ineffective testing [Source 4], accelerating time-to-market by effective testing [Source 21], and test team spending a lot of time on manual testing [Source 22]. In a presentation entitled '*Keep Calm and Use TMMi*' [Source 70], TMA/TPI is mentioned to enable accelerated time to market.

6.2.3.5. Other needs. 18 sources mentioned 'other' needs for TMA/TPI. [Source 17] argued that the adoption of automated test-case generation techniques in industry has met with some difficulty due in part to the difficulty of creating and maintaining the test models. Arising from this need, [Source 17] proposed a test process improvement model specific for automated test generation.

[Source 18] argued that a test evaluation model needs to consider embedded software characteristics and test activities in order to meet industry requirements. Arising from this need, a test process improvement model for embedded software development was proposed.

[Source 26] pointed out the lack of suitable controls of test parameters as the driver. It was argued that, without managed configurations and proper controls on setup parameters, all testing activities are futile and may lead to an endless loop of test and retest, there is a failure to directly tie the testing strategy to the organizational business needs, goals and initiatives.

Based on the need that scripted testing and exploratory testing both have weaknesses, [Source 110] proposed a hybrid testing process unifying exploratory testing and scripted testing.

6.2.4. Challenges (RQ 2.4)

Any improvement activity will come with its own challenges (impediments). After qualitative coding of the challenges (impediments) as reported in the sources, we classified them into eight categories:

1. Lack of (required) resources (mentioned in 18 sources)
2. Lack of competencies (8 sources)
3. Resistance to change (12 sources)
4. Improving feels like "an additional effort" (8 sources)
5. No clear benefits seen (4 sources)
6. Unclear scope and focus (7 sources)
7. Lack of ownership and commitment for the improvement (6 sources)
8. "Other" challenges (26 sources)

In the following, we discuss examples from each category.

6.2.4.1. Lack of resources. Many publications point out the lack of time or budget resources in general. Some sources point out more specific types of lack of resources: lack of a process improvement infrastructure [Source 19], or lack of resources especially in small and medium-sized enterprises [Source 89].

[Source 19] 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.

[Source 89], 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.

6.2.4.2. Lack of competencies. In terms of lack of competencies, [Source 32] points out that "*The staff usually has problems developing testing activities because they do not have the appropriate competences to carry out these activities effectively*" . Furthermore, [Source 82] recommended that testers should be trained to conduct TPI and improve their skills in detecting failures and writing tests.

6.2.4.3. Resistance to change. Reasons for resistance to change are for instance as follows: personal factors [Source 68], organizational factors [Source 26] as well as cultural factors [Sources 45, 106].

With regard to personal factors, [Source 68] 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.

With regard to organizational factors, [Source 26] notes that "*In some groups there is a general perception that the 'CMMI stuff' is really more for the product development groups and will not provide a direct benefit to the ATE developers*".

With regard to cultural factors, [Source 45] states that "*culture has a large role, where* e.g. *Swedish developers need to be convinced on a more personal level than more eastern cultures*", and [Source 106] recommended that TPA assessment efforts should be tailored to meet the cultural norms of the organization or there will be resistance.

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

6.2.4.4. Improving feels like an "additional effort". The subjective impression of improvement as an "additional effort" is for instance pointed out in [Source 40] noting that "*Many software developers and testers have never received training in effective software testing practices and believe that test planning is largely a waste of time*". Furthermore, as a result of an empirical study, [Source 89] reported that the activities dedicated to diagnosing the current practice, in relation to the verification and validation process improvement activities, are often considered a waste of time and money and are felt like "an additional effort".

6.2.4.5. No clear benefits seen. Reported by a team of Brazilian practitioners and researchers, [Source 76] reported that small companies that take too long 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, [Source 89] 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.

6.2.4.6. Unclear scope and focus. In terms of unclear scope and focus, [Source 47] 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. Furthermore, [Source 48] points out that "*test teams have unclear goals*".

6.2.4.7. Lack of ownership and commitment for the improvement. [Source 51] explicitly points out the lack of ownership and commitment. Furthermore, [Source 37] highlights the importance the importance of "educating" managers such that "*managers support and understand the value of the metrics*".

6.2.4.8. Other challenges. 26 sources mentioned 'other' challenges for TMA/TPI. [Source 25] points out the lack of data by reporting that "*Detailed requirements and testing processes were not described in our R&D organization*" as well as a lack of standardization in terms of testing and requirements definition by reporting that "*R&D projects applied requirements and testing practices in different ways,* i.e.*, no systematic and common practices were in use*". Furthermore, [Source 83] mentions knowledge transfer as a challenge.

[Source 35] points out the difficulty to select and adopt a specific model by reporting that "*After considering to improve its testing process, an organization faces the challenge of choosing a particular improvement model which is compatible with its goals, needs, and other constraint*" .

[Source 77] points out the problem of many items to improve, i.e., "*When many actions are suggested, applying all of these actions to the organizations reduces the effectiveness of process improvement activities, and it may lead to failure of these activities*" . Therefore a model has to be light as stated in [Source 76].

6.2.5. Benefits (RQ 2.5)

The successful implementation of TMA/TPI heavily depends on expected or actual benefits for a team, unit or organization. After qualitative coding of the benefits as reported in the sources, we classified them into three categories:

1. Business (economic) benefits (mentioned in 27 sources)
2. Operational benefits (48 sources)
3. Technical benefits (37 sources)

In the following, we discuss examples from each category.

6.2.5.1. Business (economic) benefits. Examples for business (economic) benefits of TMA/TPI are increased profit [Source 62], increased customer satisfaction [Source 62], positive return on investment [Sources 41, 70], reduced cost of test tasks [Source 50], reduction of defect costs [Sources 81, 85], better internal and external reputation [Source 100], increased business opportunities [Source 100], as well as reducing support cost [Source 114].

To show increased profit and customer satisfactions of test process improvement, [Source 62] provides a benefits chain for value-based testing process implementation.

With regard to a positive return on investment, [Source 70] points out a pay-off within 12 months. [Source 50] points out a reduced cost of test tasks by reporting that "*Based on some studies it has been concluded that the organizations at level 1 may spend $1000 for any particular task then for the same task organization at level 5 needs to spend $10*".

With regard to reduction of defect costs, [Source 85] points out a reduction of defect resolution cost by 63%. [Source 100] points out better internal and external reputation as well as increased business opportunities based on the application of TMMi. Finally, [Source 114] points out reduced support cost by 21%.

6.2.5.2. Operational benefits. Examples for operational benefits of TMA/TPI are shorter development time [Source 87], lower development cost [Source 87], better planning of testing costs [Source 55], alignment of internal testing processes with external value objectives [Source 62], better adherence to release dates [Source 63], reduced failure administration [Source 45], minimized test cycle time [Source 73], more effective and better risk identification and management [Source 100], adequate trainings for test personnel [Source 32], as well as process control based on metrics [Source 112] resulting in more accurate estimations and predictions [Source 10].

[Source 87] highlights shorter development time and lower development cost based on optimized and quantitatively managed software testing process (OptimalSQM).

[Source 55] points out better planning of testing costs as impact of the organizational model on testing, [Source 62] mentions alignment of internal testing processes with external value objectives as benefit of improving software testing process feature prioritization, and [Source 63] better adherence to release dates when improving software testing via the Orthogonal Defect Classification (ODC).

[Source 45] mentions reduced failure administration with Software Quality Rank (SQR), [Source 73] points out minimized test cycle time as a possible benefit of TMA/TPI and [Source 100] highlights more effective and better risk identification and management with TPI.

[Source 32] mentions adequate trainings for test personnel as benefit of a competence model for testing teams, [Source 112] mentions process control based on metrics as a benefit in a trans-Ireland software process improvement network, and [Source 10] highlights more accurate estimations and predictions when stating that "*We can conclude that the improved state model of the STP accounting for learning presents a major increase in the accuracy of the predictions*" .

6.2.5.3. Technical benefits. Examples for technical benefits of TMA/TPI are a reduced number of field defects resulting in a more stable product and better software quality in general [Source 63], reduction of high severity defects [Source 114], increased traceability to support release decisions [Source 67], improved test automation [Source 84], as well as improved test design by adoption of new techniques [Source 92].

[Source 63] points out better software quality in general which results in a more stable product and a reduced number of field

defects. [Source 114] refines this issue by pointing out the reduction of high severity defects in context of the model Software Test Enhancement Paradigm (STEP). [Source 67] highlights increased traceability to support release decisions in context of risk-based testing. [Source 84] and [Source 92] address benefits with regard to specific test phases, i.e., test automation and test design, respectively.

6.2.6. Methods for TPI (RQ 2.6)

With regard to methods used for TPI, the sources were classified into five categories:

1. Using a maturity or TPI model (mentioned in 93 sources)
2. Using metrics (29 sources)
3. Using control theory (4 sources)
4. Using simulation (4 sources)
5. "Other" techniques (30 sources)

In the following, we discuss the types of methods for TPI in more detail.

6.2.6.1. Maturity or TPI model. Overall 93 sources (almost 60% of all methods reported in the sources) use one or more of the reported 58 TMA/TPI models (see Section 6.1) for test maturity assessment or test process improvement. Using maturity or TPI models is therefore the most common method for TPI. For instance, [Source 56] reports on the usage of TMMi and [Source 90] on the usage of TPI.

6.2.6.2. Metrics. Overall 29 sources use metrics (independently of TMA/TPI models) as method for TPI. For instance, [Source 46] presents the test metrics model of Fabasoft which comprises the metrics Test Cases Coverage (TCC), Number of Test Cases Failed (TCF) and Passed (TCP), Number Of Test Cases Clarified (CTC), Number Of Test Cases Runs (TCR), Number Of Detected Defects (NDD), Defect Density (DD), Number of Implementation Requests (NIR), Number Of Implementation Orders (NIO), Rework Effort (RE), as well as the Total Staff Level.

6.2.6.3. Control theory. Overall 4 sources apply control theory as a method for TPI. Control theory is a branch of engineering and mathematics that deals with the behavior of dynamical systems with inputs, and how their behavior is modified by feedback. For instance, [Source 2] states "*A quantitative, adaptive process control technique is described using an industrially validated model of the software system test phase (STP) as the concrete target to be controlled. The technique combines the use of parameter correction and Model Predictive Control to overcome the problems induced by modeling errors, parameter estimation errors, and limits on the resources available for productivity improvement*". Furthermore, [Source 13] proposes a software test process stochastic control model based on CMM characterization.

6.2.6.4. Simulation. Overall 4 sources perform test process simulation as a method for TPI. Simulation is often performed to evaluate formalized TMA/TPI models. For instance, [Source 13] performs simulation to indicate the applicability of the proposed stochastic state model. Furthermore, [Source 16] validates the model MND-TMM (Ministry of National Defense – Test Maturity Model) based on Monte Carlo simulation.

6.2.6.5. Other methods. Finally, 30 sources mention other methods for TPI. For instance, [Source 48] mentions concrete guidelines, i.e., 'Know Your Efficiency', 'Institute Risk-Based Testing', 'Tighten Up Your Test Set', and 'Introduce Lightweight Test Automation'. Furthermore, [Source 62] uses value-based software engineering to improve software test processes.

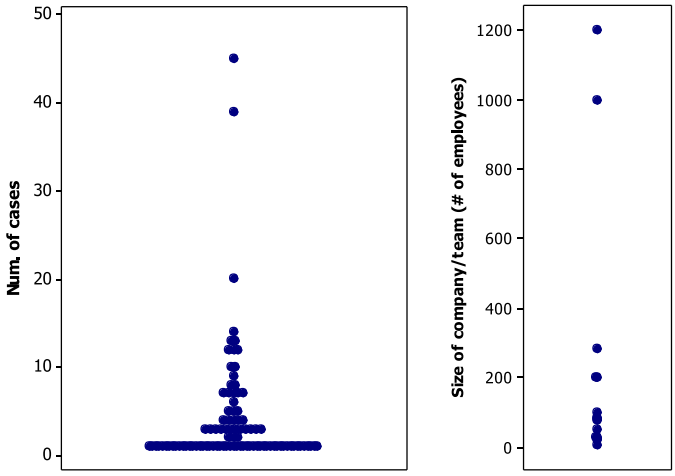

Fig. 15. Number of cases and size of company/team studied in the sources.

6.2.7. Development process models (RQ 2.7)

With regard to the applied development process model, the sources were classified into two categories: (1) plan-driven (made explicit in 14 sources), and (2) agile (made explicit in 16 sources). Furthermore, in 47 sources, the development process was not made explicit.

6.2.7.1. Plan-driven. 14 sources explicitly follow a plan-driven development process model, mainly based on the waterfall model. For instance, [Source 65] states that "*Software development follows the 'waterfall' life-cycle model*". Furthermore, [Source 81] states "*We consider functional testing for a system test phase in a project with waterfall development, experienced builders and a structured test approach driven by risk*". But also the application of other plan-driven development process models like the German V-Model XT are reported [Source 46].

6.2.7.2. Agile. 16 sources explicitly follow an agile development process model. For instance, TestSPICE is considered in the context of agile development in two sources [Sources 102, 103]. [Source 93] reflects on test process improvement in the context of agile development in general, and [Source 82] addresses the simultaneous improvement of quality and time-to-market in agile development.

6.2.8. Attributes of case-study companies / projects under study (RQ 2.8)

Fig. 15 shows the number of cases and size of company/team studied in the sources. Overall 94 of the 181 sources (51.9%) report cases including companies / projects under study. 85 sources report at most 10 cases, 51 sources report exactly one case, and 12 sources report exactly three cases. Outliers with regard to the number of cases are [Source 98, Source 31], and [Source 66], which present results based on the investigation of 20, 39, and 45 cases, respectively. In 15 sources the size of the involved company or team in terms of the number of employees is reported. This size ranges between 1 and 1200 employees.

Similar to our previous SM/SLR studies, e.g., [45–49], we also classified each source based on the type of its case-study/ projects under study, which could be either: (1) commercial or real examples (cases), and (2) hypothetical cases or simple (toy) examples made in academic settings. Judging this classification was not challenging as we could quite easily decide the type by reviewing the explanation of the case(s)/example(s) in each source. For example, [Source 96] provided a hypothetical (real-looking) example of a test suite for the insurance domain, and was classified under the type #2 above. On the other hand, [Source 105] evaluated

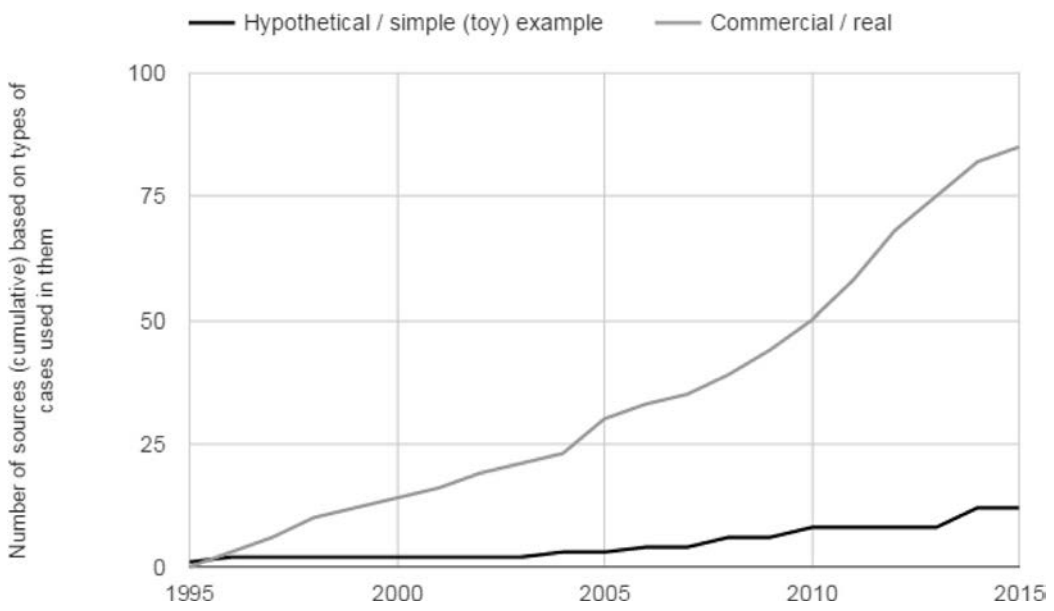

Fig. 16. Cumulative growth of the TMA/TPI field in terms of number of sources and number of commercial/real cases studied over the years.

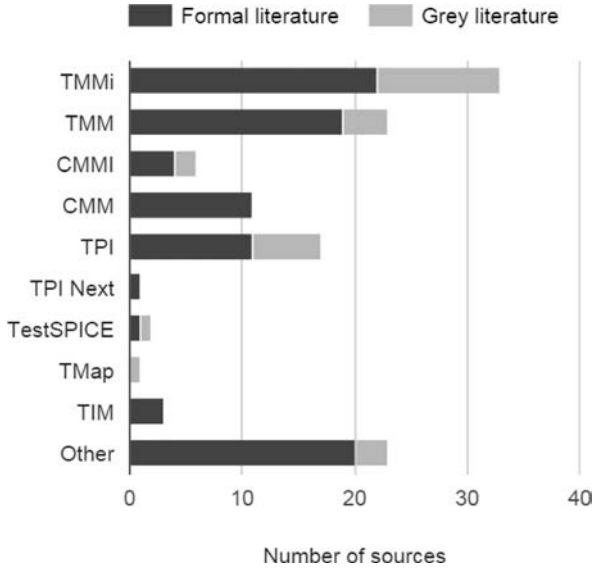

Fig. 17. Number of formal and grey literature sources per TMA/TPI model.

its proposed software test improvement model "in practice" in the context of a mid-size Finnish company.

Fig. 16 shows the cumulative number of sources based on types of cases used in them over the years. We can see that, interestingly, the number of sources using commercial or real examples (cases) has grown faster than the number of sources using hypothetical cases or simple (toy) examples.

6.3. Group 3- RQs about trends and demographics

6.3.1. Attention level in the formal and grey literature (RQ 3.1)

TMA/TPI is a field of both scientific and practical interest. This is the main reason why we performed a MLR which also includes grey literature mainly produced by practitioners. Fig. 17, which shows the number of formal and grey literature sources per TMA/TPI model, indicates that we would have missed information from practice, if we were to exclude the grey literature sources. As mentioned in Section 6.2.1, overall 58 different TMA/TPI models were identified among the sources. From these sources, 14 were grey literature reporting test maturity models such as TMap, Agile TMM or Test Maturity Index which would have been lost in a regular SLR (by not including the grey literature). Without grey literature, the usage of TMap and some other models would not have been considered.

Fig. 18 shows the attention of the TMA/TPI field over the years. The first publication on TPA/TPI appeared in 1988 and was written by practitioner's which shows that TMA/TPI is a field of research driven by practical needs. Over the years the number of publications on TMA/TPI increased but the ratio of academic, industrial and collaborative papers stayed relatively constant over the years. TMA/TPI is therefore a field which equally received attention form industry and academia.

7. Discussion

7.1. Summary of research findings and implications

We summarize the research findings of each RQ and discuss the implications next.

7.1.1. Group 1-Common to all SM studies

- **RQ 1.1: Mapping of studies by contribution facet:** We observed that there was a fine mix of contribution types in different sources. 58 sources contributed TMA/TPI models. 38 sources contributed methods, methodologies, techniques, or approaches. 6 sources contributed tool support. 13 and 18 sources contributed metrics and processes, respectively. 44 contributed empirical results mainly. 28 and 20 contributed informal heuristics / guidelines, and 'Other' types of contributions, respectively.
- **RQ 1.2: Mapping of studies by research facet:** A large number of sources (63 of them, 34%) were classified as 'weak' empirical studies which is a quite good indication of reasonable research rigor in this area of software testing, even given the fact that 51 sources in our pool (29%) were form the grey literature, which generally do not use rigorous research approaches.

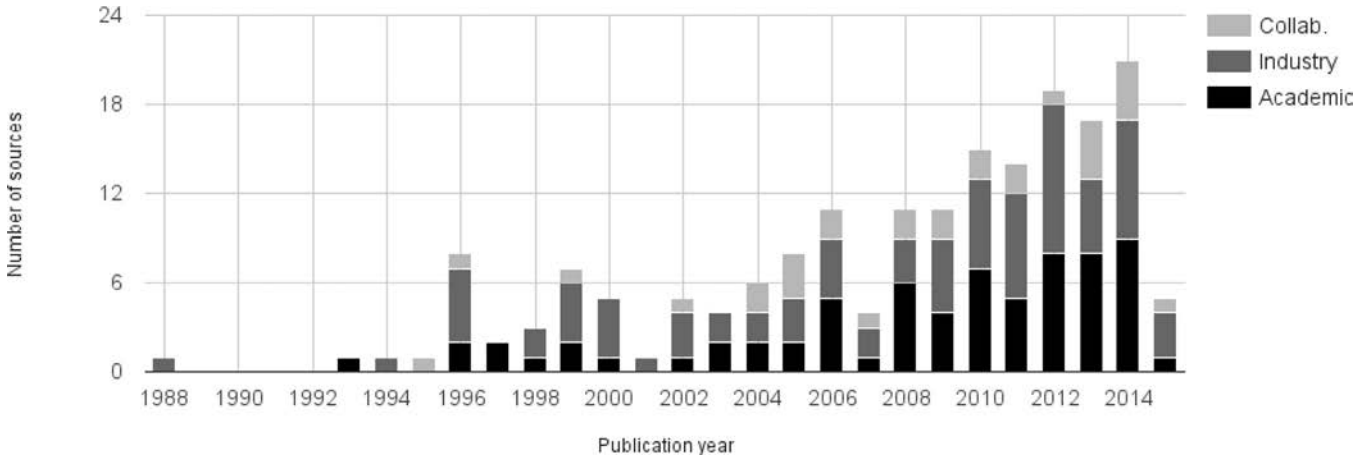

Fig. 18. Growth of the TMA/TPI field and affiliation types of the sources across different years.

7.1.2. Group 2-Specific to the domain (TMA/TPI)

- **RQ 2.1-Proposed test maturity models:** 58 of the 181 sources presented new TMA/TPI models. Being able to see 58 test maturity models out there was quite surprising to us. After reviewing the technical details of several models, authors observed that clearly many aspects in various models overlap. What is 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 the 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 need, but rather on hypothetically argued motivations and also often by not fully reviewing what is already out there to ensure minimizing the overlap.
- **RQ 2.2- Source maturity model used:** We observed that 117 sources used the existing models for TMA/TPI purposes to build newer (better or more specific) models. TMMi and its earlier version TMM were used the highest, in 34 and 23 sources, respectively. After TMMi and TMM, TPI was the 3rd most used model in the rank.
- **RQ 2.3-Drivers:** We observed that, 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 qualitative coding of the drivers as reported in the sources, we synthesized and classified drivers (needs) for TMA/TPI into five categories: Process and operational needs (mentioned in 48 sources), Needs related to software quality (25 sources), Cost-related needs (23 sources), Time and schedule-related needs (12 sources), "Other" needs (18 sources).
- **RQ 2.4-Challenges:** We observed that, as expected, any improvement activity comes with its own challenges (impediments). After qualitative coding of the challenges (impediments) as reported in the sources, we classified them into eight categories: (1) Lack of (required) resources (mentioned in 18 sources), (2) Lack of competencies (8 sources), (3) Resistance to change (12 sources), (4) Improving feels like "an additional effort" (8 sources), (5) No clear benefits seen (4 sources), (6) Unclear scope and focus (7 sources), (7) Lack of ownership and commitment for the improvement (6 sources), and (8) "Other" challenges (26 sources). Only if and when a given team can overcome the challenges, the TMA/TPI activities will be conducted with high quality and will yield benefits.
- **RQ 2.5-Benefits:** The successful implementation of TMA/TPI heavily depends on expected or actual benefits for a team, unit or organization. After qualitative coding of the benefits as reported in the sources, we classified them into three categories: Business (economic) benefits (mentioned in 27 sources), Operational benefits (48 sources), and Technical benefits (37 sources).
- **RQ 2.6-Methods for TPI:** With regard to methods used for TPI, the sources were classified into five categories: Using a maturity or TPI model (mentioned in 93 sources), Using metrics (29 sources), Using control theory (4 sources), Using simulation (4 sources), and "Other" techniques (30 sources).
- **RQ 2.7-Development process models:** Sources were classified into two categories: (1) plan-driven (made explicit in 14 sources), and (2) agile (made explicit in 16 sources). Furthermore, in 47 sources, the development process was not made explicit.
- **RQ 2.8-Attributes of case-study companies / projects under study:** Overall 94 of the 181 sources (51.9%) report cases including companies / projects under study. 85 sources report at most 10 cases, 51 sources report exactly one case, and 12 sources report exactly three cases.

7.1.3. Group 3-Trends and demographics

- **RQ 3.1-Attention level in the formal and grey literature:** TMA/TPI is a field of both scientific and practical interest. This is the main reason why we performed a MLR which also includes grey literature mainly produced by practitioners. In the finalized pool of 181 sources, 130 (71%) were formally published sources (e.g., conference and journal papers) and 51 (29%) were sources in the grey literature (e.g., internet articles and white papers).

7.2. Comparing the findings of this review with the results of the earlier reviews

Let us recall from Section 2.3 when we used Table 1 to show a summary and classification on the existing review studies in the area of TMA/TPI and the relationships, novelty and advantages of our MLR compared to them.

For the issue of comparing the findings of this review with the results of the earlier reviews, since the aspects covered in earlier reviews were quite different than our RQs (see Table 1), we cannot compare much of the findings, except the coverage in the number of the models reviewed in the studies. As we saw in Table 1, the number of TMA/TPI models reviewed in this MLR (58 models) has been more than all the previous studies (between 2 and 23 models). Furthermore, as also shown in Table 1, this MLR reviewed, synthesized and offered other aspects of this large domain

that have not been reviewed by the earlier review studies, e.g.,, we systematically synthesized and presented the drivers (needs) for TMA/TPI, challenges, benefits, and methods for TPI.

Furthermore, as reported in a recent study [11], when review studies such as SLRs do not include the grey literature in their review, they miss to review and collect important knowledge and 'state of the practice' in a given area. However, including grey literature in review studies can give substantial benefits in certain areas of SE. Thus by including the grey literature in the area of TMA/TPI, this MLR hopes to provide a more 'complete' picture of both the 'state of the art and practice' (see the results for RQ 3.1 in Section 6.3).

In summary, as discussed above, our MLR is in fact the most comprehensive review study in this domain up to now.

7.3. Potential threats to validity

The main issues related to threats to validity of this SM review are inaccuracy of data extraction, and incomplete set of studies in our pool due to limitation of search terms, selection of academic search engines, and researcher bias with regards to exclusion/inclusion criteria. In the this section, these threats are discussed in the context of the four types of threats to validity based on a standard checklist for validity threats presented in [67].

7.3.1. Internal validity

The systematic approach that has been utilized for article selection is described in Section 4. In order to make sure that this review is repeatable, search engines, search terms and inclusion/exclusion criteria are carefully defined and reported. Problematic issues in selection process are limitation of search terms and search engines, and bias in applying exclusion/inclusion criteria.

Limitation of search terms and search engines can lead to incomplete set of primary sources. Different terms have been used by different authors to point to a similar concept. In order to mitigate risk of finding all relevant studies, formal searching using defined keywords has been done followed by manual search in references of initial pool and in web pages of active researchers in our field of study. For controlling threats due to search engines, not only we have included comprehensive academic databases such as Google Scholar. Therefore, we believe that adequate and inclusive basis has been collected for this study and if there is any missing publication, the rate will be negligible.

Applying inclusion/exclusion criteria can suffer from researchers' judgment and experience. Personal bias could be introduced during this process. To minimize this type of bias, joint voting is applied in article selection and only articles with high score are selected for this study.

7.3.2. Construct validity

Construct validities are concerned with issues that to what extent the object of study truly represents theory behind the study [67]. Threats related to this type of validity in this study were suitability of RQs and categorization scheme used for the data extraction. To limit construct threats in this study, GQM approach is used to preserve the tractability between research goal and questions.

7.3.3. Conclusion validity

Conclusion validity of a SM study provided when correct conclusion reached through rigorous and repeatable treatment. In order to ensure reliability of our treatments, an acceptable size of primary sources is selected and terminology in defined schema reviewed by authors to avoid any ambiguity. All primary sources are reviewed by at least two authors to mitigate bias in data extraction. Each disagreement between authors was resolved by consensus among researchers. Following the systematic approach and described procedure ensured replicability of this study and assured that results of similar study will not have major deviations from our classification decisions.

7.3.4. External validity

External validity is concerned with to what extent the results of our multivocal literature review can be generalized. As described in Section 4, we included scientific and grey literature in the scope of test maturity assessment or test process improvement with a sound validation written in English. The issue lies in whether our selected works can represent all types of literature in the area of TMA/TPI. For this issue, we argue that relevant literature we selected in our pool taking scientific and grey literature into account contained sufficient information to represent the knowledge reported by previous researchers or professionals. As it can be seen from Section 6.3.1, the collected primary studies contain a significant proportion of academic, industrial and collaborative work which forms an adequate basis for concluding results useful for academia and applicable in industry. Also, note that our findings in this study are mainly within the field of TMA/TPI. Beyond this field, we had no intention to generalize our results. Therefore, few problems with external validity are worthy of substantial attention.

8. Conclusions and future work

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 and presented in this article 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 white papers). 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. Our article aims to benefit the readers (both practitioners and researchers) in providing the most comprehensive survey of the area, to this date, in assessing and improving the maturity of test processes.

We observed that 117 sources used the existing models for TMA/TPI purposes to build newer (better or more specific) models. TMMi and its earlier version TMM were used the highest, in 34 and 23 sources, respectively. After TMMi and TMM, TPI was the 3rd most used model in the rank. 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.

We also observed that, as expected, any improvement activity comes with its own challenges (impediments). By synthesizing the challenges reported in the sources, we classified them into eight categories: (1) lack of (required) resources, (2) lack of competencies, (3) resistance to change, (4) improving feels like "an additional effort", (5) no clear benefits seen, (6) unclear scope and focus, (7) lack of ownership and commitment for the improvement, and (8) "other" challenges. Only if and when a given team can overcome the challenges, the TMA/TPI activities will be conducted with high quality and will yield benefits. Last but not the least, by synthesizing the benefits reported in the sources, we classified them into three categories: business (economic) benefits, operational benefits, and technical benefits.

Our future work includes using the findings of this MLR in our industry-academia collaborative projects and empirical evaluation of models and techniques in the area of test maturity assessment and test process improvement as reported in this article. We also would like to explore and investigate the effectiveness of the proposed TMA/TPI models and approaches in the empirical studies,

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

and to assess the usefulness the proposed models based on the reported case studies.

Acknowledgements

Vahid Garousi was partially supported by several internal grants by Hacettepe University and the Scientific and Technological Research Council of Turkey (TÜBİTAK) via grant #115E805.

References

Sources reviewed in the MLR

[Source 1] M. Levinson, 11 Ways to Improve Software Testing, 2005 http://www.cio.com/article/2448106/developer/11-ways-to-improve-software-testing.html Last accessed: Feb. 2016.
[Source 2] S.D. Miller, R.A. DeCarlo, A.P. Mathur, J.W. Cangussu, A control-theoretic approach to the management of the software system test phase, J. Syst. Softw. 79 (2006) 1486–1503.
[Source 3] G. Hongying, Y. Cheng, A customizable agile software quality assurance model, in: Information Science and Service Science (NISS), 2011 5th International Conference on New Trends in, 2011, pp. 382–387.
[Source 4] A.F. Araújo, C.L. Rodrigues, A.M. Vincenzi, C.G. Camilo, A.F. Silva, A framework for maturity assessment in software testing for small and medium-sized enterprises, in: Proceedings of the International Conference on Software Engineering Research and Practice (SERP), 2013, p. 1.
[Source 5] M.H. Krause, A Maturity Model for Automated Software Testing, 1994 http://www.mddionline.com/article/software-maturity-model-automated-software-testing Last accessed: Nov. 2015.
[Source 6] A. Farooq, K. Georgieva, R.R. Dumke, A meta-measurement approach for software test processes, in: Multitopic Conference, 2008. INMIC 2008. IEEE International, 2008, pp. 333–338.
[Source 7] D. Karlström, P. Runeson, S. Norden, A minimal test practice framework for emerging software organizations, Softw. Test. Verif. Reliab. 15 (2005) 145–166.
[Source 8] I. Burnstein, A. Homyen, R. Grom, C. Carlson, A model to assess testing process maturity, Crosstalk 11 (1998) 26–30.
[Source 9] M. Felderer, R. Ramler, A multiple case study on risk-based testing in industry, Int. J. Softw. Tools Technol. Transf. 16 (2014) 609–625.
[Source 10] G. Abu, J.A.W. Cangussu, J. Turi, A quantitative learning model for software test process, in: System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on, 2005 78b-78b.
[Source 11] J. Kasurinen, P. Runeson, L. Riungu, K. Smolander, A self-assessment framework for finding improvement objectives with ISO/IEC 29119 test standard, in: Systems, Software and Service Process Improvement, Springer, 2011, pp. 25–36.
[Source 12] T. Suwannasart, P. Srichaivattana, A Set of Measurements to Improve Software Testing Process, 1999.
[Source 13] J.W. Cangussu, A software test process stochastic control model based on cmm characterization, Softw. Process 9 (2004) 55–66.
[Source 14] H. Ryu, D.-K. Ryu, J. Baik, A strategic test process improvement approach using an ontological description for mnd-tmm, in: Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on, 2008, pp. 561–566.
[Source 15] J. Park, H. Ryu, H.-J. Choi, D.-K. Ryu, A survey on software test maturity in Korean defense industry, in: Proceedings of the 1st India Software Engineering Conference, 2008, pp. 149–150.
[Source 16] J. Park, A Test Maturity Model to Improve Test Process in Developing Defense System Software MSc thesis, School of Engineering, Information and Communications University, Korea, 2008.
[Source 17] H. Heiskanen, M. Maunumaa, M. Katara, A test process improvement model for automated test generation, in: Product-Focused Software Process Improvement, Springer, 2012, pp. 17–31.
[Source 18] E. Jung, A test process improvement model for embedded software developments, in: Quality Software, 2009. QSIC'09. 9th International Conference on, 2009, pp. 432–437.
[Source 19] D. McAndrews, J.M. Ryan, P. Fowler, A Turbo-Team Approach to Establishing a Software Test Process at Union Switch and Signal, 1997 DTIC Document.
[Source 20] C. Lee, Adapting and adjusting test process reflecting characteristics of embedded software and industrial properties based on referential models, in: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, 2009, pp. 1372–1377.
[Source 21] S.R. Shanmuga Karthikeyan, Adopting the Right Software Test Maturity Assessment Model, 2014 http://www.cognizant.com/InsightsWhitepapers/Adopting-the-Right-Software-Test-Maturity-Assessment-Model-codex881.pdf Last accessed: Dec.2015.
[Source 22] P. Eshtiagh, An Evaluation of Test Processes in an Agile Environment, 2014.
[Source 23] P.M. Bueno, A.N. Crespo, M. Jino, Analysis of an artifact oriented test process model and of testing aspects of CMMI, in: Product-Focused Software Process Improvement, Springer, 2006, pp. 263–277.
[Source 24] A. Kasoju, K. Petersen, M.V. Mäntylä, Analyzing an automotive testing process with evidence-based software engineering, Inf. Softw. Technol. 55 (2013) 1237–1259.
[Source 25] J. Kukkanen, K. Vakevainen, M. Kauppinen, E. Uusitalo, Applying a systematic approach to link requirements and testing: a case study, in: Software Engineering Conference, 2009. APSEC'09. Asia-Pacific, 2009, pp. 482–488.
[Source 26] S.T. Stevens, Applying CMMI and strategy to ATE development, in: Autotestcon, 2006 IEEE, 2006, pp. 813–818.
[Source 27] L. Lazić, D. Velašević, Applying simulate on and design of experiments to the embedded software testing process, Softw. Test. Verif. Reliab. 14 (2004) 257–282.
[Source 28] F. Duncan, A. Smeaton, Assessing and improving software quality in safety critical systems by the application of a SOFTWARE TEST MATURITY MODEL, 7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012, 2012.
[Source 29] S. Ronen- Harel, ATMM Agile Testing Maturity Model, 2010 http://www.slideshare.net/AgileSparks/atmm-practical-view Last accessed: Feb. 2016.
[Source 30] A. Rodrigues, A. Bessa, P.R. Pinheiro, Barriers to implement test process in small-sized companies„ in: Organizational, Business, and Technological Aspects of the Knowledge Society, Springer, 2010, pp. 233–242.
[Source 31] K.G. Camargo, F.C. Ferrari, S.C. Fabbri, Characterising the state of the practice in software testing through a TMMi-based process, J. Softw. Eng. Res. Develop. 3 (2015) 1–24.
[Source 32] J. Saldaña-Ramos, A. Sanz-Esteban, J. García-Guzmán, A. Amescua, Design of a competence model for testing teams, IET Softw. 6 (2012) 405–415.
[Source 33] I. Burnstein, T. Suwanassart, R. Carlson, Developing a testing maturity model for software test process evaluation and improvement, in: Test Conference, 1996. Proceedings., International, 1996, pp. 581–589.
[Source 34] I.S. Burnstein, Taratip, C.R. Carlson, Developing a Testing Maturity Model: Part I, 1996 http://www.elen.ktu.lt/~rsei/PT/Developing%20a%20Testing%20Maturity%20Model%20Part%201%20-%20Aug%201996.htm Last accessed: Dec.2015.
[Source 35] A. Farooq, R.R. Dumke, Developing and applying a consolidated evaluation framework to analyze test process improvement approaches, in: Software Process and Product Measurement, Springer, 2008, pp. 114–128.
[Source 36] J. Jacobs, Development & validation of a metric based test maturity model, in: EUROSTAR 2001: Proceedings of the 9th European International Conference on Software Testing Analysis& Review, 2001.
[Source 37] Y. Chen, R.L. Probert, K. Robeson, Effective test metrics for test strategy evolution, in: Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, 2004, pp. 111–123.
[Source 38] J. Kasurinen, Elaborating software test processes and strategies, in: Software Testing, Verification and Validation (ICST), 2010 Third International Conference on, 2010, pp. 355–358.
[Source 39] H. Kumar, N. Chauhan, Emergence of testing maturity model, in: National Indial Conference INDIACOM, 2009, p. 3.
[Source 40] G. Daich, Emphasizing software test process improvement, Crosstalk (1996) 20–26.
[Source 41] Accenture, Enterprise Test Assessments – The Who, What, When, Why and How, 2012 http://cqaa.org/Resources/Documents/Presentations%202012/Enterprise%20Test%20Assessments%20Overview%20Half%20Day%20-%20Allstate%20QAI%202012.pdf Last accessed: Feb. 2016.
[Source 42] L.-O. Damm, Evaluating and Improving Test Efficiency MSE-2002-15. Master's thesis, 2002.
[Source 43] S. Alone, K. Glocksien, Evaluation of Test Process Improvement Approaches: An Industrial case Study, 2015.
[Source 44] F. O'Hara, Experiences from informal test process assessments in Ireland–Top 10 findings, in: Software Process Improvement and Capability Determination, Springer, 2011, pp. 194–196.
[Source 45] S. Eldh, S. Punnekkat, H. Hansson, Experiments with Component Tests to Improve Software Quality, 2007.
[Source 46] N. Hrgarek, Fabasoft best practices and test metrics model„ J. Inf. Organ. Sci. 31 (2007) 75–89.
[Source 47] L.O. Damm, L. Lundberg, C. Wohlin, Faults-slip-through—a concept for measuring the efficiency of the test process, Softw. Process 11 (2006) 47–59.
[Source 48] R. Black, Four Ideas for Improving Test Efficiency, 2008 http://www.rbcs-us.com/images/documents/Four-Ideas-for-Improving-Software-Test-Efficiency.pdf Last accessed: Feb. 2016.
[Source 49] J. Kasurinen, O. Taipale, K. Smolander, How test organizations adopt new testing practices and methods? in: Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on, 2011, pp. 553–558.
[Source 50] M. Balajiwale, How to Achieve Level 5 Maturity for QA and Testing Process, November 2015 2015.
[Source 51] G. Ottosen, How to Eat an Elephant: Tips for Facilitating Test Process Improvement, 2015 Last accessed: Feb. 2016.
[Source 52] L.-O. Damm, L. Lundberg, Identification of test process improvements by combining fault trigger classification and faults-slip-through measurement, in: Empirical Software Engineering, 2005. 2005 International Symposium on, 2005, p. 10.

[Source 53] K. Gomes Camargo, F. Cutigi Ferrari, S. Camargo Pinto Ferraz Fabbri, Identifying a subset of TMMi practices to establish a streamlined software testing process, in: Software Engineering (SBES), 2013 27th Brazilian Symposium on, 2013, pp. 137–146.
[Source 54] T. Toroi, A. Raninen, L. Vaatainen, Identifying process improvement targets in test processes: a case study, in: Software Maintenance (ICSM), 2013 29th IEEE International Conference on, 2013, pp. 11–19.
[Source 55] J.J. Ahonen, T. Junttila, M. Sakkinen, Impacts of the organizational model on testing: three industrial cases, Empir. Softw. Eng. 9 (2004) 275–296.
[Source 56] G. Thompson, Implementing Test Maturity Model Integrated (TMMi) Workshop, 2009 http://www.bcs.org/upload/pdf/gthompson1-190912.pdf Last accessed: Feb. 2016.
[Source 57] H. Germundsson, Improvement Areas for Agile Test Processes, 2012 http://www.diva-portal.org/smash/get/diva2:555780/FULLTEXT01.pdf Last accessed: Feb. 2016.
[Source 58] T. Koomen, M. Pol, Improvement of the test process using TPI, in: Proc. Sogeti Nederland BV, 1998.
[Source 59] Q. Li, B. Boehm, Improving scenario testing process by adding value-based prioritization: an industrial case study, in: Proceedings of the 2013 International Conference on Software and System Process, 2013, pp. 78–87.
[Source 60] H. Fujita, M. Mejri, Improving software test processes, in: New Trends in Software Methodologies, Tools and Techniques: Proceedings of the Fifth SoMeT 06, 2006, p. 462.
[Source 61] O. Taipale, K. Smolander, Improving software testing by observing practice, in: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, 2006, pp. 262–271.
[Source 62] Q. Li, Y. Yang, M. Li, Q. Wang, B.W. Boehm, C. Hu, Improving software testing process: feature prioritization to make winners of success-critical stakeholders, J. Softw. 24 (2012) 783–801.
[Source 63] M. Butcher, H. Munro, T. Kratschmer, Improving software testing via ODC: three case studies, IBM Syst. J. 41 (2002) 31–44.
[Source 64] K. Kim, R.Y.C. Kim, Improving test process for test organization assessed with TMMi based on TPI NEXT, Int. J. Softw. Eng. Appl. 8 (2014) 59–66.
[Source 65] H.K. Leung, Improving the testing process based upon standards, Softw. Test. Verif. Reliab. 7 (1997) 3–18.
[Source 66] M. Hoppe, A. Engel, Improving VVT process in SysTest project: evaluating the results of pilot projects in six industries, INCOSE 2005, in: 15th International Symposium, Rochester, New York, USA, 2005, pp. 10–15.
[Source 67] M. Felderer, R. Ramler, Integrating risk-based testing in industrial test processes, Softw. Qual. J. 22 (2014) 543–575.
[Source 68] W.D. Woodruff, Introduction of Test Process Improvement and the Impact on the Organization, 2003.
[Source 69] ISO, ISO/IEC/ IEEE 29119-2, Software and Systems Engineering – Software Testing – Part 2: Test Processes, 2013 http://www.iso.org/iso/catalogue_detail.htm?csnumber=56736 Last accessed: Feb. 2016.
[Source 70] C. Bates, Keep Calm and Use TMMi, 2012 http://fistb.ttlry.mearra.com/sites/fistb.ttlry.mearra.com/files/Clive%20Bates%20Keep%20Calm%20and%20use%20TMMi_0.pdf Last accessed: Feb. 2016.
[Source 71] M. Steiner, M. Blaschke, M. Philipp, T. Schweigert, Make test process assessment similar to software process assessment—the Test SPICE approach, J. Softw. 24 (2012) 471–480.
[Source 72] T. Abdou, P. Grogono, P. Kamthan, Managing corrective actions to closure in open source software test process, The 25th International Conference on Software Engineering and Knowledge Engineering (SEKE 2013), 2013.
[Source 73] R.F. Goldsmith, Managing, Measuring and Improving the Testing Processes, Software Quality Group of New England (SQGNE), 2012 Last accessed: Feb. 2016.
[Source 74] P. Kapur, G. Singh, N. Sachdeva, A. Tickoo, Measuring software testing efficiency using two-way assessment technique, in: Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2014 3rd International Conference on, 2014, pp. 1–6.
[Source 75] J. Gorin, "Methods and Apparatus for test Process Enhancement," ed: Google Patents, 2003.
[Source 76] A.P.C.C. Furtado, M.A.W. Gomes, E.C. Andrade, I.H. De Farias Jr, MPT. BR: a Brazilian maturity model for testing, in: Quality Software (QSIC), 2012 12th International Conference on, 2012, pp. 220–229.
[Source 77] H. Oh, B. Choi, H. Han, W.E. Wong, Optimizing test process action plans by blending testing maturity model and design of experiments, in: Quality Software, 2008. QSIC'08. The Eighth International Conference on, 2008, pp. 57–66.
[Source 78] S. Morasca, D. Taibi, D. Tosi, OSS-TMM: guidelines for improving the testing process of open source software, Int. J. Open Source Softw. Process. 3 (2011) 1–22.
[Source 79] A. Raninen, Brunou Paivi, Practical Process Improvements – Applying the LAPPI Technique in QA and Testing, 2014 http://profes2014.cs.helsinki.fi/wp-content/uploads/2013/09/Tutorial_Raninen_Brunou_LAPPI.pdf Last accessed: Feb. 2016.
[Source 80] V.N. Maurya, D.K. Arora, A.K. Maurya, D. Singh, A. Priyadarshi, G. Anand, et al., Product Quality Improvement and Project Assessment using Test Maturity Model Integrated (TMMi), 2013.
[Source 81] M. Ruiz, P.J. Tuya González, D. Crespo, Simulation-based optimization for software dynamic testing processes, Int. J. Adv. Softw. 7 (1-2) (2014).
[Source 82] P.M. Dóra, A.C. Oliveira, J.A.B. Moura, Simultaneously improving quality and time-to-market in agile development, in: Software Technologies, Springer, 2014, pp. 84–98.
[Source 83] J. Kasurinen, Software organizations and test process development, Adv. Comput. 85 (2012) 1–63.
[Source 84] Spire, Software Process Improvement in Regions of Europe (SPIRE)-Boom, 1998 http://www.cse.dcu.ie/cse_www/pdf/publication/spire/boomen.pdf Last accessed: Feb. 2016.
[Source 85] J. Lee, S. Hwang, Software test capability improvement method, in: Computer Applications for Software Engineering, Disaster Recovery, and Business Continuity, Springer, 2012, pp. 246–251.
[Source 86] H. Fliek and S. Christensen, "Software Testing Capability Assessment Framework," ed: Google Patents, 2014.
[Source 87] L. Lazić, Software testing optimization by advanced quantitative defect management, Comput. Sci. Inf. Syst. 7 (2010) 459–487.
[Source 88] T. Schweigert, A. Nehfort, Technical issues in test process assessment and their current handling in TestSPICE, J. Softw. 26 (2014) 350–356.
[Source 89] J. García, A. de Amescua, M. Velasco, A. Sanz, Ten factors that impede improvement of verification and validation processes in software intensive organizations, Softw. Process 13 (2008) 335–343.
[Source 90] Capgemini, Test Maturity Assessment and Improvement using TPI® and Quality Blueprint, 2010 https://www.capgemini.com/resource-file-access/resource/pdf/test_maturity_assessment_and_improvement_using_tpi_and_quality_blueprint.pdf Last accessed: Feb. 2016.
[Source 91] T. Foundation, Test Maturity Model integration (TMMi), 2012 http://www.tmmi.org/pdf/TMMi.Framework.pdf Last accessed: Feb. 2016.
[Source 92] P. Gerrard, Test Process Improvement, 2000 http://gerrardconsulting.com/sites/default/files/ProcImp60mins.pdf Last accessed: Dec, 2015.
[Source 93] E. van Veenendaal, Test Process Improvement and Agile: Friends or Foes?, 2014 http://www.erikvanveenendaal.nl/NL/files/testingexperience27_09_14_van_Veenendaal.pdf Last accessed: Feb. 2016.
[Source 94] G. Thompson, Test Process Improvement and the Test Manager: An Uncomfortable Marriage, 2008 http://www.powershow.com/view/1432d9-MzdlM/Test_Process_Improvement_and_the_Test_Manager_An_uncomfortable_marriage_powerpoint_ppt_presentation Last accessed: Feb. 2016.
[Source 95] E. van Veenendaal, R. Grooff, R. Hendriks, Test Process Improvement using TMMi, 2007 http://improveqs.nl/files/Test_Process_Improvement_using_TMMi_STAR_Tester_2007-10.pdf Last accessed: Feb. 2016.
[Source 96] F. Haser, M. Felderer, R. Breu, Test process improvement with documentation driven integration testing, in: Quality of Information and Communications Technology (QUATIC), 2014 9th International Conference on the, 2014, pp. 156–161.
[Source 97] N. Barrett, S. Martin, C. Dislis, Test process optimization: closing the gap in the defect spectrum, in: Null, 1999, p. 124.
[Source 98] J.J. Cannegieter, Tester, Get Out of Your Cave!, 2011 http://www.bcs.org/upload/pdf/jjcannegieter-131212-1.pdf.
[Source 99] T. Schweigert, D. Vohwinkel, M. Blaschke, M. Ekssir-Monfared, Test-SPICE and agile testing–synergy or confusion, in: Software Process Improvement and Capability Determination, Springer, 2013, pp. 154–164.
[Source 100] M. Gebauer Dunlop, The Benefits of Test Process Improvement, 2012 https://www.youtube.com/watch?v=lZ2cwp1OLbE Last accessed: Feb. 2016.
[Source 101] T. Schweigert, A. Nehfort, M. Ekssir-Monfared, The feature set of Test-SPICE 3.0, in: Systems, Software and Services Process Improvement, Springer, 2014, pp. 309–316.
[Source 102] M. Heusser, The Fishing Maturity Model, 2010 http://www.stickyminds.com/article/fishing-maturity-model Last accessed: Feb. 2016.
[Source 103] S. Reid, The new software testing standard, in: C. Dale, T. Anderson (Eds.), Achieving Systems Safety, Springer, London, 2012, pp. 237–255.
[Source 104] S. Reid, The personal test maturity matrix, in: CAST 2006: Influencing the Practice June 5th-7th, 2006–Indianapolis, 2006, p. 133.
[Source 105] L. Zhang, The Software Test Improvement Model in Practice, 2005.
[Source 106] I. Burnstein, The testing maturity model and test process assessment, in: Practical Software Testing: A Process-Oriented Approach, 2003, pp. 537–585.
[Source 107] M.D. Karr, The Unit Test Maturity Model, 2013 http://davidmichaelkarr.blogspot.com.tr/2013/01/the-unit-test-maturity-model.html Last accessed: Feb. 2016.
[Source 108] G. d. Vries, The What and How of Testing-TMap Next and TPI Next, 2010 https://www.scribd.com/doc/180328914/TMap-Next-and-TPI-Next-Related Last accessed: Feb. 2016.
[Source 109] M. Miller, A. Goslin, I. Test-UK, TMM-a case study, Software Quality Conferences incorporating ICSTEST, 2005.
[Source 110] S.M.A. Shah, C. Gencel, U.S. Alvi, K. Petersen, Towards a hybrid testing process unifying exploratory testing and scripted testing, J. Softw. 26 (2014) 220–250.

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

[Source 111] Sogeti, TPI Automative, Verison 1.01, 2004 http://www.tpiautomotive.de/Docs/TPI%20automotive%20version%201.01.pdf Last accessed: Feb. 2016.
[Source 112] Nnorthen Telecom, Trans-Ireland Software Process Improvement Network (TRI-SPIN)-NORTHERN TELECOM, 1996 http://www.cse.dcu.ie/cse/international/trispin/nortel.html Last accessed: Feb. 2016.
[Source 113] W. Yaddow, Using Test Maturity Models and Industry QA Standards for Test Process Improvement, 2006 http://www.stickyminds.com/article/using-test-maturity-models-and-industry-standards-test-process-improvement.
[Source 114] P. Chelladurai, Watch Your STEP, 2011 http://www.uploads.pnsqc.org/2011/papers/T-56_Chelladurai_paper.pdf Last accessed: Feb. 2016.
[Source 115] D. Gelperin, in: What's Your Testability Maturity?, Application Trends, 1996, pp. 50–53.

Other references

[1] T. Britton, L. Jeng, G. Carver, P. Cheak, T. Katzenellenbogen, Reversible Debugging Software, University of Cambridge, Judge Business School, 2013 *Tehnical Report*.
[2] V. Garousi, A. Coşkunçay, A.B. Can, O. Demirörs, A survey of software engineering practices in Turkey, J. Syst. Softw. 108 (2015) 148–177.
[3] V. Garousi, J. Zhi, A survey of software testing practices in Canada,, J. Syst. Softw. 86 (2013) 1354–1376.
[4] M. Grindal, J. Offutt, J. Mellin, On the testing maturity of software producing organizations, Testing: Academia & Industry Conference-Practice And Research Techniques, 2006.
[5] O. Taipale, K. Smolander, Improving software testing by observing practice, in: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, 2006, pp. 262–271.
[6] G. Bath, E.V. Veenendaal, Improving the Test Process: Implementing Improvement and Change - A Study Guide for the ISTQB Expert Level Module, Rocky Nook, 2013.
[7] E. Tom, A. Aurum, R. Vidgen, An exploration of technical debt, J. Syst. Softw. 86 (6) (2013) 1498–1516.
[8] R.T. Ogawa, B. Malen, Towards rigor in reviews of multivocal literatures: applying the exploratory case study method, Rev. Educ. Res. 61 (1991) 265–286.
[9] M.Q. Patton, Towards utility in reviews of multivocal literatures, Rev. Educ. Res. 61 (1991) 287–292.
[10] V. Garousi, M.V. Mäntylä, When and what to automate in software testing? A multi-vocal literature review, Inf. Softw. Technol. 76 (2016) 92–117.
[11] V. Garousi, M. Felderer, M.V. Mäntylä, The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature, in: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, 2016, p. 26.
[12] W. Afzal, S. Alone, K. Glocksien, R. Torkar, Software test process improvement approaches: a systematic literature review and an industrial case study, J. Syst. Softw. 111 (2016) 1–33.
[13] C. Garcia, A. Dávila, M. Pessoa, Test process models: systematic literature review, in: A. Mitasiunas, T. Rout, R. O'Connor, A. Dorling (Eds.), Software Process Improvement and Capability Determination, vol. 477, Springer International Publishing, 2014, pp. 84–93.
[14] P. Chandra, Software Assurance Maturity Model, A Guide to Building Security into Software Development, Version 1.0, 2009 http://www.opensamm.org/downloads/SAMM-1.0.pdf Last accessed: Oct. 2016.
[15] S. Kollanus, ICMM—a maturity model for software inspections, . Softw. Maint. Evol. 23 (2011) 327–341.
[16] E.v. Veenendaal, B. Wells, Test Maturity Model integration (TMMi): Guidelines for Test Process Improvement, Uitgeverij Tutein Nolthenius, 2012.
[17] TMMI Foundation, TMMI Specification (reference model), release 1.0, Oct. 2015 http://www.tmmi.org/pdf/TMMi.Framework.pdf Last accessed.
[18] T. Koomen, M. Pol, Test Process Improvement: A Practical Step-By-Step Guide to Structured Testing, Addison-Wesley, 1999.
[19] A. v. Ewijk, B. Linker, M. v. Oosterwijk, B. Visser, TPI Next: Business Driven Test Process Improvement, Kleine Uil, 2013.
[20] I. Burnstein, A. Homyen, R. Grom, and C.R. Carlson, "A model to assess testing process maturity," *Crosstalk* vol. 11, 1998.
[21] A. Ampatzoglou, A. Ampatzoglou, A. Chatzigeorgiou, P. Avgeriou, The financial aspect of managing technical debt: a systematic literature review, Inf. Softw. Technol. 64 (8) (2015) 52–73.
[22] R.L. Glass, T. DeMarco, Software Creativity 2.0, developer.* Books, 2006.
[23] W.F. Whyte, Participatory Action Research, SAGE Publications, 1990.
[24] K.M. Benzies, S. Premji, K.A. Hayden, K. Serrett, State-of-the-evidence reviews: advantages and challenges of including grey literature,, Worldviews Evid. Based Nurs. 3 (2006) 55–61.
[25] Q. Mahood, D. Van Eerd, E. Irvin, Searching for grey literature for systematic reviews: challenges and benefits, Res. Synth. Methods 5 (2014) 221–234.
[26] S. Hopewell, M. Clarke, S. Mallett, Grey literature and systematic reviews, in: H.R. Rothstein, A.J. Sutton, M. Borenstein (Eds.), Publication Bias in Meta-Analysis: Prevention, Assessment and Adjustments, John Wiley & Sons, 2006.
[27] S. Hopewell, S. McDonald, M. Clarke, M. Egger, Grey literature in meta-analyses of randomized trials of health care interventions, Cochrane Database Systematic Rev. (2007).
[28] I. Kulesovs, iOS Applications Testing, in: iOS Applications Testing, 3, 2015, pp. 138–150.
[29] M. Sulayman, E. Mendes, A systematic literature review of software process improvement in small and medium web companies, in: D. Ślęzak, T.-h. Kim, A. Kiumi, T. Jiang, J. Verner, S. Abrahão (Eds.), Advances in Software Engineering, 59, Springer Berlin Heidelberg, 2009, pp. 1–8.
[30] A. Yasin, M.I. Hasnain, On the Quality of Grey Literature and its Use in Information Synthesis During Systematic Literature Reviews Master Thesis, Blekinge Institute of Technology, Sweden, 2012.
[31] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Evidence-Based Software Engineering, 2007 Evidence-Based Software Engineering.
[32] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: an update, Inf. Softw. Technol. 64 (2015) 1–18.
[33] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, Presented at the International Conference on Evaluation and Assessment in Software Engineering (EASE), 2008.
[34] R.J. Adams, P. Smart, A.S. Huff, Shades of grey: guidelines for working with the grey literature in systematic reviews for management and organizational studies, Int. J. Manag. Revi. (2016) n/a-n/a.
[35] J. Tyndall, AACODS Checklist for Evaluation and Critical Appraisal of Grey Literature, Flinders University, 2010 http://canberra.libguides.com/content.php?pid=660700&sid=5471876 Last accessed: Nov. 2016.
[36] A.R. Swinkels, A Comparison of TMM and other Test Process Improvement, Company Frits Philips Institute, 2000 white paper https://wwwbruegge.informatik.tu-muenchen.de/lehrstuhl_1/files/teaching/ws0708/ManagementSoftwareTesting/12-4-1-FPdef.pdf Last accessed: Dec. 2015.
[37] A. Farooq, R.R. Dumke, Research directions in verification & validation process improvement, SIGSOFT Softw. Eng. Notes, 32 (2007) 3.
[38] Maveric Systems Co., Test Process Improvement - Evaluation of Available Models, Dec. 2015 white paper http://maveric-systems.com/pdf/whitepapers/White_paper_Vol%208_TPI.pdf Last accessed:.
[39] Cognizant Co., Adopting the Right Software Test Maturity Assessment Model, Company Frits Philips Institute, 2014 white paper http://www.cognizant.com/InsightsWhitepapers/Adopting-the-Right-Software-Test-Maturity-Assessment-Model-codex881.pdf Last accessed: Dec. 2015.
[40] A. Farooq, Evaluation Approaches in Software Testing, Universität Magdeburg, Germany, 2008 *Technical report, FIN-05-2008*.
[41] A. Farooq, An Evaluation Framework for Software Test Processes PhD thesis, Universität Magdeburg, Germany, 2009.
[42] C.G. v. Wangenheim, J.C.R. Hauck, C.F. Salviano, A. v. Wangenheim, Systematic literature review of software process capability/maturity models, in: Proceedings of International Conference on Software Process. Improvement And Capability dEtermination (SPICE), 2010.
[43] K. Rungi, R. Matulevičius, Empirical analysis of the Test Maturity Model integration (TMMi), in: T. Skersys, R. Butleris, R. Butkiene (Eds.), Information and Software Technologies, vol. 403, Springer Berlin Heidelberg, 2013, pp. 376–391.
[44] T. Abdou, P. Grogono, P. Kamthan, Managing corrective actions to closure in open source software test process, International Conference on Software Engineering and Knowledge Engineering, 2013.
[45] J. Zhi, V. Garousi, B. Sun, G. Garousi, S. Shahnewaz, G. Ruhe, Cost, benefits and quality of software development documentation: a systematic mapping, J. Syst. Softw. 99 (2015) 175–198.
[46] V. Garousi, Y. Amannejad, A. Betin-Can, Software test-code engineering: a systematic mapping, J. Inf. Softw. Technol. 58 (2015) 123–147.
[47] S. Doğan, A. Betin-Can, V. Garousi, Web application testing: a systematic literature review, J. Syst. Softw. 91 (2014) 174–201.
[48] F. Häser, M. Felderer, R. Breu, Software paradigms, assessment types and non-functional requirements in model-based integration testing: a systematic literature review, in: Presented at the Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, 2014.
[49] M. Felderer, P. Zech, R. Breu, M. Büchler, A. Pretschner, Model-based security testing: a taxonomy and systematic classification,, Softw. Test. Verif. Reliab. (2015).
[50] M. Banks, Blog posts and tweets: the next frontier for grey literature, in: D. Farace, J. Schöpfel (Eds.), Grey Literature in Library and Information Studies, Walter de Gruyter, 2010.
[51] K. Godin, J. Stapleton, S.I. Kirkpatrick, R.M. Hanning, S.T. Leatherdale, Applying systematic review search methods to the grey literature: a case study examining guidelines for school-based breakfast programs in Canada, Syst. Rev. 4 (Oct 22 2015) 138.

[52] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Presented at the Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, England, United Kingdom, 2014.
[53] V. Garousi, M. Felderer, T. Hacaloğlu, Online Paper Repository for the MLR on 'Software Test Maturity Assessment and Test Process Improvement', Jan. 2016 http://goo.gl/zwY1sj Last accessed.
[54] D.S. Cruzes, T. Dybå, Synthesizing evidence in software engineering research, in: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010.
[55] D.S. Cruzes, T. Dybå, Recommended steps for thematic synthesis in software engineering, in: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 275–284.
[56] D.S. Cruzesa, T. Dybåb, Research synthesis in software engineering: a tertiary study, Inf. Softw. Technol. 53 (2011) 440–455.
[57] G.S. Waliaa, J.C. Carverb, A systematic literature review to identify and classify software requirement errors, Inf. Softw. Technol. 51 (2009) 1087–1109.
[58] S. Ali, L.C. Briand, H. Hemmati, R.K. Panesar-Walawege, A systematic review of the application and empirical investigation of searchbased test case generation, IEEE Trans. Softw. Eng. 36 (2010) 742–762.
[59] H. Cooper, L.V. Hedges, J.C. Valentine, The Handbook of Research Synthesis and Meta-Analysis, 2nd ed., Russell Sage Foundation, 2009.
[60] M.B. Miles, A.M. Huberman, J. Saldana, Qualitative Data Analysis: A Methods Sourcebook, 3rd ed., SAGE Publications Inc., 2014.
[61] V. Garousi, A. Mesbah, A. Betin-Can, S. Mirshokraie, A systematic mapping of web application testing, Inf. Softw. Technol. 55 (8) (2013) 1374–1396.
[62] I. Banerjee, B. Nguyen, V. Garousi, A. Memon, Graphical User Interface (GUI) testing: systematic mapping and repository, Inf. Softw. Technol. 55 (10) (2013) 1679–1694.
[63] N. Bencomo, S. Hallsteinsen, E. Santana de Almeida, A view of the dynamic software product line landscape, IEEE Comput. 45 (2012) 36–41.
[64] S. Cepeda, CMMI - Staged or Continuous?, 2005 https://www.sei.cmu.edu/library/assets/cepeda-cmmi.pdf Last accessed: Nov. 2015.
[65] O. Güngör, V. Garousi, A. Kapakli, K. Herkiloğlu, Application of the TMMI and TPI-Next Frameworks: An Industrial Case Study in the Defense Sector, 2015 Paper in preparation.
[66] M. Felderer, R. Ramler, Integrating risk-based testing in industrial test processes„ Softw. Qual. J. 22 (2014) 543–575.
[67] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, 2000.

Further reading

L. Lian, F. Fujita, S. Kusumoto, K.-i. Matsumoto, T. Kikuno, K. Torii, A model-based approach for software test process improvement, in: Software Quality and Productivity, Springer, 1995, pp. 300–307.
P. Cruz, R. Villarroel, F. Mancilla, M. Visconti, A software testing process for the reference model of Competisoft, in: Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the, 2010, pp. 51–59.
V. Kettunen, J. Kasurinen, O. Taipale, K. Smolander, A study on agility and testing processes in software organizations, in: Proceedings of the 19th International Symposium on Software Testing and Analysis, 2010, pp. 231–240.
I. Burnstein, G. Saxena, R. Grom, A Testing Maturity Model for Software Test Process Assessment and Improvement, 1999.
J. Andrade, J. Ares, M.-A. Martínez, J. Pazos, S. Rodríguez, J. Romera, et al., An architectural model for software testing lesson learned systems, Inf. Softw. Technol. 55 (2013) 18–34.
A. Farooq, An Evaluation Framework for Software Test Processes, Otto von Guericke University Magdeburg, 2009.
R. Alchorne, R. Nóbrega, G. Aquino, S.R. Meira, A. Vasconcelos, Balanced testing scorecard-a model for evaluating and improving the performance of software testing houses, in: ICEIS, 1, 2010, pp. 370–378.
K.K. Rana, S.S.U. Ahmad, Bringing maturity to test [Source software testing], Electron. Syste. Softw. 3 (2005) 32–35.
A. Farooq, K. Georgieva, R.R. Dumke, Challenges in evaluating SOA test processes, in: Software Process and Product Measurement, Springer, 2008, pp. 107–113.
I. Burnstein, Controlling and monitoring the testing process, in: Practical Software Testing: A Process-Oriented Approach, 2003, pp. 263–302.
A. Black, Critical Testing Process: Plan, Prepare, Perform, Perfect, Addison-Wesley Longman Publishing Co., Inc., 2003.
I.C. Burnstein, C.R. Carlson, Developing a Testing Maturity Model, Part II, 1993 http://kopustas.elen.ktu.lt/~rsei/PT/Developing%20a%20Testing%20Maturity%20Model,%20Part%20II.htm Last accessed: Feb. 2016.
E. Lamas, L.A.V. Dias, A.M. da Cunha, Effectively testing for a software product line with OTM3 organizational testing management maturity model, in: Information Technology: New Generations (ITNG), 2011 Eighth International Conference on, 2011, pp. 274–279.
K. Rungi, R. Matulevičius, Empirical analysis of the test maturity model integration (TMMi), in: Information and Software Technologies, vol. 403, 2013, pp. 376–391.
L. Fernández-Sanz, M.T. Villalba, J.R. Hilera, R. Lacuesta, Factors with negative influence on software testing practice in spain: a survey, in: Software Process Improvement, Springer, 2009, pp. 1–12.
Codenomicon, Fuzzy Testing Maturity Model, 2014 http://www.codenomicon.com/resources/white-paper/2013/11/01/fuzz-maturity-model.html.
G. Bath, E.V. Veenendaal, Improving the Test Process: Implementing Improvement and Change - A Study Guide for the ISTQB Expert Level Module, Rocky Nook, 2013.
P. Belt, Improving Verification and Validation Activities in ICT Companies–Product Development Management Approach, Improving Verification and Validation Activities in ICT Companies–Product Development Management Approach, vol. 324, Acta Universitatis Ouluensis C Technica, 2009.
E. Dustin, Improving Your Software Testing: Focus on the Testing Team, 2003 http://www.informit.com/articles/article.aspx?p=31196 Last accessed: Feb. 2016.
S.H. Kan, J. Parrish, D. Manlove, In-process metrics for software testing, IBM Syst. J. 40 (2001) 220–241.
W. Afzal, R. Torkar, Incorporating metrics in an organizational test strategy, in: in Software Testing Verification and Validation Workshop, 2008. ICSTW'08. IEEE International Conference on, 2008, pp. 304–315.
A. Tarhan, O. Demirors, Investigating the effect of variations in the test development process: a case from a safety-critical system, Softw. Qual. J. 19 (2011) 615–642.
R. Black, Managing the Testing Process, John Wiley & Sons, 2002.
A. Systems, Measure Your Test Maturity Index, 2011 http://www.aspiresys.com/testmaturityindex/ Last accessed: Feb. 2016.
Y. He, H. Hecht, R. Paul, Measuring and assessing software test processes using test data, in: High Assurance Systems Engineering, 2000, Fifth IEEE International Symposim on. HASE 2000, 2000, pp. 259–264.
RBCS, Metrics for Software Testing: Managing with Facts: Part 2: Process Metrics, 2011 Last accessed: Feb. 2016.
M. Grindal, J. Offutt, J. Mellin, On the testing maturity of software producing organizations, in: Testing: Academic and Industrial Conference-Practice And Research Techniques, 2006. TAIC PART 2006. Proceedings, 2006, pp. 171–180.
E. Lamas, É. Ferreira, M.R. do Nascimento, L.A.V. Dias, F.F. Silveira, Organizational testing management maturity model for a software product line, in: Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, 2010, pp. 1026–1031.
E. Lindahl, Pimp My Test Process: Introducing Test Automation and Process Maturity in an IT Consulting Context, 2012.
L. Howard, Process Improvement Reviews, 2014 https://www.planittesting.co.nz/resource/process-improvement-reviews/ Last accessed: Feb. 2016.
Accenture, Process Improvement with TMMi, 2010 http://www.tmmi.org/pdf/Presentation.Accenture.2010.pdf Last accessed: Feb. 2016.
Spire, Software Process Improvement in Regions of Europe (SPIRE)-AD Ricera, 1998 http://www.cse.dcu.ie/cse_www/pdf/publication/spire/aden.pdf Last accessed: Feb. 2016.
Spire, Software Process Improvement in Regions of Europe (SPIRE)-Boom, 1998 http://www.cse.dcu.ie/cse_www/pdf/publication/spire/advent.pdf Last accessed: Feb. 2016.
S. Sudarsanam, Software test process assessment methodology, J. Eng. Comput. Appl. Sci. 2 (2013) 52–56.
Deloitte, Test Maturity Model Assessment Helping Clients, 2009.
Experimentus, Test Maturity Model Integrated (TMMi)- How Mature are Companies Software Quality Management Processes in Today's Market, 2011 http://www.experimentus.com/wp-content/uploads/2015/08/experimentustmmisurveyresults.pdf Last accessed: Feb. 2016.
E.C. van Veenendaal, Jan Jaap, Test Maturity Model integration (TMMi) Results of the First TMMi Benchmark – Where are We Today?, 2012 http://www.tmmi.org/pdf/e-book_tmmi.pdf Last accessed: Feb. 2016.
A. Systems, Test Maturity Tweens, Teens and Twenties (Webinar by Aspire Systems), 2011 http://softwareengi.com/scrum_checklist/Test_Maturity___Tweens__Teens__Twenties_/770467/.
G.C. Limited, Test Organisation Maturity Questionnaire V2.0, 2006 http://testassurance.com/tom/tom200.pdf Last accessed: Feb. 2016.
T. Aaltio, Test Proces Improvement with TPI Next: What the Model Does Not Tell you but you Should Know, 2013 http://www.slideshare.net/VLDCORP/test-process-improvement-with-tpi-next-what-the-model-does-not-tell-you-but-you-should-know Last accessed: Feb. 2016.
Lionbridge, Test Process Assessments: Move into The Real World, 2009 http://whitepapers.venturebeat.com/whitepaper7264 Last accessed: Feb. 2016.
J. Söderlund, M. Lindgren, D. Sundmark, Test Process Improvement & Test/Build Tool Evaluation Master thesis, Mälardalen University, Sweden, 2010.
H. Heiskanen, M. Maunumaa, M. Katara, Test Process Improvement for Automated Test Generation, Tampere University of Technology, Department of Software Systems, Tampere, 2010.
P. Walen, Test Process Improvement: Lessons Learned From the Trenches, 2011 http://www.softwaretestpro.com/Item/5063/Test-Process-Improvement-Lessons-Learned-From-the-Trenches/Software-Test-Professionals-Conference-Testing-Process-Management-Forums-Software Last accessed: Feb. 2016.
S. Kulkarni, Test process maturity models–yesterday, today and tomorrow, in: Proceedings of the 6th Annual International Software Testing Conference, Delhi, India, 2006.
C. Garcia, A. Dávila, M. Pessoa, Test process models: systematic literature review, in: Software Process Improvement and Capability Determination, Springer, 2014, pp. 84–93.

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001

D. Gelperin, B. Hetzel, The growth of software testing, in: Communications of the ACM, 31, 1988, pp. 687–695.
J. Jacobs, J. van Moll, T. Stokes, The process of test process improvement, XOOTIC Mag. 8 (2000) 23–29.
S.C. Reid, The software component testing standard (BS 7925-2), in: Quality Software, 2000. Proceedings. First Asia-Pacific Conference on, 2000, pp. 139–148.
T. Ericson, A. Subotic, S. Ursing, TIM - a test improvement model, Softw. Test. Verif. Reliab. 7 (1997) 229–246.
T. Tayamanon, T. Suwannasart, N. Wongchingchai, A. Methawachananont, TMM appraisal assistant tool, in: Systems Engineering (ICSEng), 2011 21st International Conference on, 2011, pp. 329–333.
Experimentus, TMMi Survey 2012 Update, 2012 http://www.tmmi.org/pdf/tmmisurvey2012.pdf Last accessed: Feb. 2016.
Experimentus, TMMi Survey 2013, 2013 https://www.youtube.com/watch?v=VGSx_agcbHQ Last accessed: Feb. 2016.
A.P.C.C. Furtado, S.R.L. Meira, M.W. Gomes, Towards a maturity model in software testing automation, in: Ninth International Conference on Software Engineering Advances, 2014, pp. 282–285.
J. Jacobs, J.J. Trienekens, Towards a metrics based verification and validation maturity model, in: Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop on, 2002, pp. 123–128.
S. Eldh, K. Andersson, A. Ermedahl, K. Wiklund, Towards a test automation improvement model (TAIM), in: Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on, 2014, pp. 337–342.
K. Smilgyte, R. Butleris, Towards software testing process improvement from requirements, in: Information and Software Technologies, Springer, 2012, pp. 474–481.
E. Computing, Trans-Ireland Software Process Improvement Network (TRI-SPIN)-ESBI Computing, 1996 http://www.cse.dcu.ie/cse/international/trispin/esbi.html Last accessed: Feb. 2016.
PACE, Trans-Ireland Software Process Improvement Network (TRI-SPIN)-PACE, 1996 http://www.cse.dcu.ie/cse/international/trispin/pace.html Last accessed: Feb. 2016.
D.F. Rico, Using Cost Benefit Analyses to Develop Software Process Improvement (SPI) Strategies, 2000 Contract Number SP0700-98-D-4000, AFRL/IF, DACS, Rome, NY.
M. Felderer, A. Beer, Using defect taxonomies for testing requirements, Softw. IEEE 32 (2015) 94–101.
M. Felderer, A. Beer, Using defect taxonomies to improve the maturity of the system test process: results from an industrial case study, in: Software Quality. Increasing Value in Software and Systems Development, Springer, 2013, pp. 125–146.
T.C. Staab, Using SW-TMM to improve the testing process, Crosstalk J. Def. Softw. Eng. 15 (2002) 13–17.
J. Sirathienchai, P. Sophatsathit, D. Dechawatanapaisal, Using test employee capability maturity model for supporting gaps bridging in software testing, J. Softw. Eng. Appl. 5 (2012) 417.
K. Olsen, P.S. Vinje, Using the testing maturity model SM in practical test-planning and post-evaluation, in: Proc. 6th European Software Testing, Analysis and Review Conference (EuroSTAR), Munich, 1998, pp. 345–359.
VeriTest, Veritest Test Maturity Assessment, 2014 http://www.veritest.com/testing-services/test-maturity/vtma/.

Please cite this article as: V. Garousi et al., Software test maturity assessment and test process improvement: A multivocal literature review, Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.01.001