DOI: 10.46541/978-86-7233-386-2_11

25th International Scientific Conference
Strategic Management and Decision Support Systems
in Strategic Management

19th May, 2020, Subotica, Republic of Serbia

Sara Gračić
University of Novi Sad, Faculty of Economics
Subotica, Serbia
saritta4u@gmail.com

MINIMIZING ACADEMIA VS. INDUSTRY CONFLICT THROUGH IDENTIFICATION OF COMMON TEST MODELS' PROCESS AREAS AND MATURITY LEVELS' VARIATIONS

Abstract: The quality of developed software is highly influenced by the quality of the development process. Testing process, as a stage in software development, especially, contributes to software quality. However, testing process requires significant effort and has been in the focus of both academia and practice. As a result, many test process models have been developed. Problems arise, because of gap between academia and practice. Some models are just on a conceptual level, others do not take into account needs of the industry; some models are general, while others are domain specific. One research has also shown that many models lack assessment tools. The main objective of this paper is to minimize the gap between industry and academia by comparing test process models and determining common test process activities and variations in maturity models. To accomplish this, three scientific databases were chosen for conducting primary research: Springer Link, Science Direct and IEEE xplore. Search string structured in the form of "test process model" was applied in the above stated databases. Springer Link search resulted in 129 hits; Science Direct search gave 139 hits, while IEEE xplore search resulted in 15 hits. One of the resulting papers reported 23 test process models and standards, so secondary search was conducted as well. All available models (from primary and secondary search), were then compared. Maturity levels and process areas were extracted. Differences in maturity levels, as well as common activities for available test process models have been identified and presented, with the purpose of bridging the gap between industry and academia and thus increasing the quality of testing processes in software organizations, since higher quality of testing will lead to higher quality software.

Keywords: test process models, maturity models, process areas, test process improvement

1. INTRODUCTION

Software quality is strongly influenced by the quality of the development process and testing process, especially, contributes to software quality. However, test process requires significant effort (Garcia, Dávila, & Pessoa, 2014). Since software evolves, test cases break and are discarded. This is a "waste of effort and leads to test suites that are less effective and have lower coverage". Test repair approaches have been developed to eliminate this problem (Imtiaz, Sherin, Khan, & Iqbal, 2019). "Test smells", as poorly designed tests, also waste effort, so various guidelines for their prevention, detection and correction are being offered (Garousi, & Küçük, 2018). Embedded software, due to a vast number of connections, requires effective and efficient testing, for which many approaches, tools, techniques and frameworks have

How to cite: Gračić, S. (2020). Minimizing Academia Vs. Industry Conflict Through Identification of Common Test Models' Process Areas and Maturity Levels' Variations. In *Proceedings of the 25th International Scientific Conference Strategic Management and Decision Support Systems in Strategic Management.* https://doi.org/10.46541/978-86-7233-386-2_11

been proposed by practitioners and researchers (Garousi, Felderer, Karapıcak, & Yılmaz, 2018). To advance testing process and thus, the quality of developed software, a lot of test process models and test process improvement models have been developed over the years. This, however, created problems, because of gap between academia and industry. Some developed models are just on a conceptual level, others do not take into account real needs of the industry; some models are general, while others are domain specific. Many models also lack assessment tools (Afzal, Alone, Glocksien, & Torkar, 2016). What could be done to minimize this gap and improve testing quality and, therefore, the quality of developed software?

After the Introduction, the rest of the paper is structured in the following order: Section 2 gives short overview of current trends in the field of test process models; Section 3 presents research methodology (research objective, questions, search string, inclusion and exclusion criteria, chosen scientific databases and process of building the final population); Section 4 extracts data regarding maturity levels' variations and common process areas; Section 5 discusses findings, while conclusion is given in Section 6.

2. BACKGROUND

Garousi, Felderer, & Hacaloglu in papers published in 2017 and 2018 reported that the most popular test maturity models were Test Maturity Model Integration (TMMi), its predecessor, Testing Maturity Model (TMM), Test Process Improvement (TPI) and its successor, TPI NEXT. They also concluded that out of 58 models, no one model fits all test process improvement needs, possibly because some models from academia are not based on real, industrial needs, but "hypothetically argued motivations", or because some researchers do not fully take into account best practices from academia and industry. Due to so many available models to choose from, selecting the most appropriate one is becoming quite difficult. Garcia et al. (2014) identified 23 test process models. They stated that many of them are just an adaptation or extension of TMMi and TPI. TMMi and TPI are considered as "mainstream" models and are in focus of other researchers. Models, such as TOM and MMAST have a low adoption rate. TMMi and TPI are maturity models. ISO/IEC-29119 in conjunction with ISO/IEC 33063 is a capability model. CMMI is the base for TMMi. TPI uses a test maturity matrix. ISO/IEC 29119 architecture is defined by Process Reference Models (PRMs) and Process Assessment Models (PAMs). 13 models they examined are general, 9 are domain specific. 17% of investigated models are based on TMM, 19% on TPI, "followed by a 12% of practical experience models". Afzal et al. (2016) compared content of TPI NEXT and TMMi. Their research has showed two important problems with many software test process improvement (STPI) approaches: insufficient information and lack of assessment instruments. It was pointed out that approximately 61% of STPI approaches have been developed as concepts, which only augments that implementation of these approaches in industry poses great difficulty. In fact, Afzal et al. (2016) argue that numerous STPI approaches are not applicable in industry, with the exception of domain specific approaches e.g. Emb-TPI. They pointed out that some aspects of level 2 maturity of TMMi are spread across all 3 maturity levels of TPI NEXT. So, if an organization achieves the highest level of TPI NEXT, it will not surpass level 2 of TMMi.

3. RESEARCH METHODOLOGY

The above stated problems focus the attention on what could be done to minimize the gap between industry and academia regarding the choice of most appropriate test process model to ensure higher quality of developed software.

3.1. Research Objective

The main objective of this paper is to summarize the existing software test process models and standards to determine variations in maturity levels and common process areas, since they are a source of confusion regarding an organization's position in terms of test process improvement and actions taken, which would lead to achieving the best practice in the software development industry. Interpretation of found variations and common process areas should help in minimizing current gap between academia and industry, which will further increase the level of understanding of existing models and thus, ensure higher level of adoption in software industry and will have positive impact on software quality.

3.2. Defining Research Questions

Based on the set objective, following research questions were constructed. RQ1: What common process areas can be identified and how are they presented in different models? RQ2: Are there inconsistencies among maturity levels and how significant are they? RQ3: Why are practitioners reluctant in accepting test process models? RQ4: How can model confusion be minimized or eliminated? RQ5: Can identified common test process areas and maturity level variations help in building more industry appropriate test process models?

3.3. Defining Search String for Scientific Databases

To provide answers to research questions, a literature review was conducted to identify existing test process models. Three electronic databases were chosen for conducting this research: IEEE xplore, Science Direct and Springer Link. Research string was set in the following way: "test process model" and was applied in all of the above stated electronic databases.

3.4. Defining Inclusion and Exclusion Criteria

There were no limitations regarding publishing period (all studies available till March 3rd 2020 were taken into consideration). All publications types were included in the initial population - journals, conferences and books. Both published and approved papers were taken into consideration. Grey literature (e.g. blog posts, white papers) did not enter the initial population. Only studies in the domain of test process models were examined. Studies that were either not focused on test process models, or could not be accessed, were excluded. Language was not set as a barrier for entering the initial population, although it could be for entering the next level (if title or abstract are not in English).

3.5. Building the Final Population

Search string was applied in the above mentioned databases. This resulted in building an initial population of total of 283 hits that were either published or approved for publishing. Hits were structured in the following order: Science Direct (139), Springer Link (129) and IEEE xplore (15). To ensure that a study is focused on test process models, titles, keywords and abstracts of all 283 papers were evaluated. It was determined that 282 articles were published in English completely; 1 was published in German and was excluded from the second population; papers that could not be accessed were also excluded. 10 research papers from IEEE xplore were included in the second population. From accessible research papers from Science Direct, only 7 were included. 15 papers from Springer Link were included in the second population. After "quick reading" of the second population, the third population was built out of 16 papers. One resulting paper pointed to several available test process models, which were also included (from formal or informal sources), so final population consists of 25 papers.

4. EXTRACTING VARIATIONS IN TEST PROCESS MODELS

To improve quality of testing process, a lot of test process models and test process improvement models, with variations in maturity levels and process areas have been developed. These variations created a vast gap between academia and industry, because they generate confusion regarding where software companies are in terms of improvement of testing processes and what actions should be taken to reach the best practice in the software development industry. Therefore, maturity levels and process areas were extracted from test process models and standards.

4.1. Maturity level variations

TMM model has 5 maturity levels: Initial, Phase-Definition, Integration, Management and measurement and Optimizing/Defect prevention and quality control (Afzal et al., 2016), (Oh, Choi, Han, & Wong, 2008), (Burnstein, Suwannasart, & Carlson, 1996). **TMMi model** has 5 maturity levels: Initial, Managed, Defined, Measured and Optimization (Afzal et al., 2016), (Garousi et al., 2017), (van Veenendaal, & Cannegieter, n.d.). **TIM model** has 5 maturity levels: Initial, Baselining, Cost-effectiveness, Risk-lowering, Optimizing (Afzal et al., 2016), (Kasurinen, Runeson, Riungu, & Smolander, 2011), (Ericson, Subotic, & Ursing, 1997). **Self-assessment framework for ISO/IEC 29119 based on TIM** has 5 maturity levels: Initial, Baseline, Cost-effectiveness, Risk-lowering and Optimization (Afzal et al., 2016), (Kasurinen, 2012). **CMM** and **CMMI** have 5 maturity levels: Initial (ad hoc and chaotic processes), Managed (requirements are managed, processes are planned and measured), Defined (processes are standardized across the organization and tailored by each project), Quantitatively managed (processes are measured and controlled using statistics) and Optimizing (processes are continually improved) (no authors, 2007), (Bueno, Crespo, & Jino, 2006).
TPI model has 3 maturity levels: Controlled, Efficient and Optimized (Afzal et al., 2016), (Garousi et al., 2017). **TPI NEXT model** has 4 maturity levels: Initial, Controlled, Efficient and Optimized (Afzal et al., 2016). **MB-VV-MM model** has 5 maturity levels: Initial, Repeatable, Defined, Managed and aligned and Optimizing (Afzal et al., 2016). **Unit Test Maturity Model** has 9 maturity levels: Ignorance, Few simple tests, Mocks and stubs, Design for testability, Test driven development, Code coverage, Unit tests in the build, Code coverage feedback loop and Automated builds and tasks (Garousi et al., 2017). **Automated Software Testing Maturity Model** has 4 maturity levels: Accidental, Beginning, Intentional and Advanced automation (Garousi et al., 2017). **Agile Quality Assurance Model (AQAM)** has 4 maturity levels: Initial, Performed, Managed and Optimized (Garousi et al., 2017). **Agile Testing Maturity Model (ATMM)** has 5 maturity levels: Waterfall, Forming, Agile bonding, Performing and Scaling (Garousi et al., 2017). **MPT.BR: A Brazilian Maturity Model for Testing** includes 5 maturity levels: Partially Managed, Managed, Defined, Defect Prevention and Automation and Optimization (Furtado, Gomes, Andrade, & de Farias, 2012). **MND-TMM model** is

designed for weapon software system development and has 5 maturity levels for each of the 10 process areas (Afzal et al., 2016), (Ryu, Ryu, & Baik, 2008).
TPI Automotive model (Afzal et al., 2016) and **ATG add-on for TPI model** (Afzal et al., 2016), (Heiskanen, Maunumaa, & Katara, 2012) have 4 maturity levels that are different for different areas. **Emb-TPI model** has maturity structure and 18 key areas (Afzal et al., 2016), (Jung, 2009). The last 3 are implemented in the area of testing embedded software. **Test SPICE model** (Afzal et al., 2016), (Steiner, Blaschke, Philipp, & Schweigert, 2012), software testing standards **ISO/IEC 29119** and **ISO 33063** (Afzal et al., 2016), (Kasurinen et al., 2011) and **MTPF (Minimal test practice framework)** do not have maturity models (Afzal et al., 2016), (Karlström, Runeson, & Nordén, 2005).

4.2. Process areas variations

TMM model includes process areas: testing and debugging, goals and policies, test planning process, testing techniques and methods, test organization, technical training program, software life cycle, controlling and monitoring, review test measurement program, software quality evaluation, defect prevention, quality control, test process optimization (Afzal et al., 2016), (Oh et al., 2008), (Burnstein et al., 1996). **TMMi's** process areas are: test policy and strategy, test planning, test monitoring and control, test design and execution, test environment, test organization, test training program, test lifecycle and integration, non-functional testing, peer reviews, test measurement, product quality evaluation, advanced reviews, defect prevention, quality control, test process optimization (Afzal et al., 2016), (Garousi et al., 2017), (van Veenendaal, & Cannegieter, n.d.).
Self-assessment framework for ISO/IEC 29119 based on TIM has following process areas: Organizational test process (OTP), Test management process (TMP), Test planning process (TPP), Test monitoring and control process (TMCP), Test completion process (TCP), Static test process (STP), Dynamic test process (DTP) (Afzal et al., 2016), (Kasurinen, 2012). **MND-TMM model** includes process areas: military (software quality evaluation), process (test strategy, test planning and test process management), infrastructure (test organization, test environment and testware management) and techniques (testing techniques, test specification, fault management) (Afzal et al., 2016), (Ryu et al., 2008). **TIM model** includes process areas: Organization, Planning and tracking, Test cases, Testware and Reviews (Afzal et al., 2016), (Kasurinen et al., 2011), (Ericson et al., 1997).
MB-VV-MM model includes process areas: V&V Environment, V&V Design methodology, V&V Monitor and control, V&V Project planning, V&V Policy and goals, Peer reviews, V&V Lifecycle embedding, Training and program, Organization embedding, Qualitative process measurement, Quality measurement and evaluation, Organizational alignment, Process optimization, Quality management, Defect prevention (Afzal et al., 2016). **TPI model** includes process areas: Test strategy, Life-cycle model, Moment of involvement, Estimation and planning, Test specification techniques, Static test techniques, Metrics, Test tools, Test environment, Office environment, Commitment and motivation, Test functions and training, Scope of methodology, Communication, Reporting, Defect management, Testware management, Test process management, Evaluation and Low-level testing (Afzal et al., 2016), (Garousi et al., 2017).
TPI NEXT model has following process areas: Stakeholder commitment, Degree of involvement, Test strategy, Test organization, Communication, Reporting, Test process management, Estimating and planning, Metrics, Defect management, Testware management, Methodology practice, Tester professionalism, Test case design, Test tools and Test environment (Afzal et al., 2016).
TPI Automotive has following process areas: Test strategy, Life-cycle model, Moment of involvement, Estimation and planning, Test design techniques, Static test techniques, Metrics, Test automation, Test environment, Office and laboratory environment, Commitment and motivation, Test functions and training, Scope of methodology, Communication, Reporting, Defect management, Testware management, Test process management, Evaluation Low-level testing and Integration testing (Afzal et al., 2016). **ATG add-on for TPI model's** process areas include: Test strategy, Life-cycle model, Moment of involvement, Estimation and planning, Test specification techniques, Static test techniques, Metrics, Test tools, Test environment, Office environment, Commitment and motivation, Test functions and training, Scope of methodology, Communication, Reporting, Defect management, Testware management, Test process management, Evaluation, Low-level testing, Modeling approach, Use of models, Test confidence, Technological and methodological knowledge (Afzal et al., 2016), (Heiskanen et al., 2012). **Emb-TPI model** includes 18 key process areas with 6 categories: Test process, Test technique, Test automation, Test quality, Test organization and Test infrastructure (Afzal et al., 2016), (Jung, 2009).
Test SPICE model process areas include: Primary life cycle processes; Test service acquisition; Test service supply; Test environment operation; Testing Supporting life cycle processes; Test process support; Organizational life cycle processes; Management Resource and infrastructure; Process improvement for test and Regression and reuse engineering (Afzal et al., 2016), (Steiner et al., 2012).
Software testing standards **ISO/IEC 29119** and **ISO 33063** process areas include: Test policy; Organizational test strategy; Test plan; Test status report; Test completion report; Test design specification; Test case specification; Test procedure specification; Test data requirements; Test environment requirements; Test data readiness report; Test environment readiness report; Test execution log and Incident report (Afzal et al., 2016), (Kasurinen et al., 2011).
MTPF (Minimal test practice framework) process areas include: Problem and experience reporting; Roles and organization issues; Verification and validation; Test administration and Test planning (Afzal et al., 2016), (Karlström et

al., 2005). **CenPRA testing process** includes process areas like: test preparation; execution and recording (Bueno et al., 2006). **V model** includes test preparation and execution (Bueno et al., 2006). **ATP model** includes processes: test planning; test design; test execution and test summary (Yang, Yuan, & Zhang, 2015). **ISTQB test process model** describes fundamental test process: planning and control; analysis and design; implementation and execution; evaluating exit criteria and reporting; and test closure activities (Sánchez-Gordón, & Colomo-Palacios, 2018). **Test generation process model** includes system specification, system configuration, test criteria definition and test generation (von Mayrhauser, Walls, & Mraz, 1994), (von Mayrhauser, & Mraz, 1995).

Key areas in **TAIM** include: *Test Management* (Planning, Deployment including Resource Management, Evaluation, Reports, Status and Progress, Automation Analysis, Technical Debt); *Test Requirements* (Specific Test Automation Requirements e.g. tools or environment, Testability, Analysis of System, Analysis of Architecture (e.g. testability, scope, dependencies/slicing – components/integration levels)); *Test Specifications* (Test Case Generation, Test Design Techniques, Pre-process analysis of test, Build in features e.g. Constraints, Properties, Invariants); *Test code – implementation* (Architecture of test code, Standards/templates/patterns, Test Code Language, Static/ Dynamic measurements on test code); *Test Automation Process* (Context, Type, Level. Process Metrics on e.g. Speed); *Test Execution* (Test Case Selection, Priority, Type of test technique (test goal),Regression Tests); *Test Verdicts* (Test Oracles, Post-Process Analysis, Results (e.g. test case verdict, logs, reports)); *Test Environment – context* (Test environment specification & set up, Type (e.g. Simulation, Emulation, Target), Test Data, Certification suites, API's); *Test Tools* (Tool Selection, Integration, (Interchange), Tool chains, Tool(s) Architecture, Frameworks, API's, Components, Installation, Upgrade, Changeability) and *Fault/ Defect handling* (Change Reports/Anomaly (Failure/Bug) Reports, Classifications, Fault identification, Triaging, Fault localizations, Fault Correction, Fault Prediction) (Eldh, Andersson, & Wiklund, 2014).

MPT.BR: A Brazilian Maturity Model for Testing process areas include: *Partially Managed* (Test Project Management, Test Project and Implementation), *Managed* (Test Project Management (evolution), Test Project and Implementation (evolution) and Test Requirement Management), *Defined* (Test Project Management (evolution), Test Project and Implementation (evolution), Test Closure, Quality Assurance, Test Measurement and Analysis, Test Organization, Acceptance Testing, Static Testing, Training), *Defect Prevention* (Test Organization (evolution), Product Quality Assessment, Management of Defects, Non-functional Testing) and *Automation and Optimization* (Automating the conduct of Tests CEP – Statistical Control of the process GDF – Management of Tools) (Furtado et al., 2012).

5. DISCUSSION

Software quality is especially determined by the quality of testing process, which requires significant effort (Garcia et al., 2014). To advance testing process, a lot of test process models have been developed and they have "overwhelmed" the testing community. This only created a gap between academia and industry, since variations in maturity levels and process areas create confusion, which makes tracking an organization's development quite difficult or even contradictory.

5.1. Interpretation of differences in maturity levels

Examined test process models were classified in 3 groups based on information found regarding maturity levels. *The first* are test process models that have **explicitly stated maturity levels**: TMM, TMMi, MB-VV-MM, TIM, TPI, TPI NEXT, Self-assessment framework for ISO/IEC 29119 based on TIM, Unit Test Maturity Model, Agile Quality Assurance Model (AQAM), Automated Software Testing Maturity Model, Agile Testing Maturity Model (ATMM), CMM, CMMI and MPT.BR-A Brazilian Maturity Model for Testing. *The second* are test process models/standards that have **no maturity model**. Those are: Test SPICE, Minimal test practice framework (MTPF), Software testing standards ISO/IEC 29119 and ISO 33063. *The third* are test process models that have **different maturity levels for different fields of application**: TPI Automotive model, Emb-TPI model, ATG add-on for TPI model and MND-TMM model.

Since organizations strive to achieve the best testing practice, they should continuously improve their testing process and reached maturity level should indicate how developed testing practice is in an observed software organization. However, software organizations are faced with a problem - which maturity model to choose. The problem becomes even more serious when there are inconsistencies among maturity models. Inconsistencies generate confusion regarding positioning in terms of test process improvement and what should be done to move to a higher maturity level.

Different maturity models assume different starting points and different criteria for assessing how mature testing processes are. Some examined models (TMM, TMMi, TIM, TPI NEXT, MB-VV-MM, CMM, CMMI, Agile Quality Assurance Model, Self-assessment framework for ISO/IEC 29119 based on TIM) assume that testing processes are ad hoc and chaotic and that the first stage of (im)maturity is "initial". Others, on the other hand, assume that organization have already partially managed testing processes (e.g. MPT.BR: A Brazilian Maturity Model for Testing) or controlled testing processes (e.g. TPI).

Majority of the observed models agree that the last stage of reaching full maturity of testing processes is when they are automated and optimized. But another problem occurs, since the meaning of "optimized" differs in different models. It has to be pointed out that for e.g. CMMi this means that testing processes are continuously improved, while for MPT.BR: A Brazilian Maturity Model for Testing this means that processes are controlled statistically, which is the fourth level of maturity in CMMi. Therefore, if an organization reaches level 5 of MPT.BR, it will be on the level 4 of CMMi. Afzal et al. (2016) also noticed that some aspects of level 2 maturity of TMMi are spread across all 3 maturity levels of TPI NEXT. So, if an organization achieves the highest level of TPI NEXT, it will not surpass level 2 of TMMi.

When it comes to defense systems, avionics, home appliances, medical systems, etc., it can be stated that maturity models have to be (and they are) tuned for a specific field and often have many maturity levels. E.g. MND-TMM model for defense has 5 maturity levels for each of the 10 process areas (Afzal et al., 2016), (Ryu et al., 2008), that is, 50 maturity levels to focus on. TPI Automotive model (Afzal et al., 2016) and ATG add-on for TPI model (Afzal et al., 2016), (Heiskanen et al., 2012) have 4 maturity levels that are different for different areas. Emb-TPI model has maturity structure and 18 key areas, which are determined for each branch of embedded software industry (Afzal et al., 2016), (Jung, 2009).

5.2. Identification of common process areas

Different test process models vary in number, names and scope of process areas, which cause problems for an organization looking to apply one of the examined models. E.g. V model has only 2 major process areas: preparation and execution, while MND-TMM model has 10.

Also, one of the most important aspects, when testing any software, is **test planning**. Many of the observed models (TMM, TMMi, Self-assessment framework for ISO/IEC 29119 based on TIM, MND-TMM, MB-VV-MM, TIM, TPI, TPI NEXT, TPI Automotive, ATG add-on for TPI, Software testing standards ISO/IEC 29119 and ISO 33063, MTPF, ATP, ISTQB, TAIM) include explicitly this process area, which cannot be said for V model and CenPRA. By looking "deeper" into the scope of CenPRA and V model's process areas, however, the first area (preparation) does include test planning. This only stresses out how important is to understand the scope of every process area of an observed model, before creating another, potentially similar, test model.

Since quality of testing process influences software/product quality, **quality** process area is also very important and is a part of various test process models, but in different forms. *Software/product* ***quality*** *evaluation/assessment* is present in TMM, TMMi, MND-TMM and MPT.BR (Brazilian Maturity Model for Testing). MPT.BR also insists on *quality assurance* and so does Test SPICE; TMM and TMMi use *quality control*; MB-VV-MM is focused on *quality management*, while Emb-TPI insists on *test quality*.

During software development, defects/bugs that are discovered during testing process have to be removed. Therefore, **defect management** plays an important role in creating high quality software. Various test process models incorporate different terms for dealing with defects. E.g. term *defect prevention* is found in TMM, TMMi, MB-VV-MM and MPT.BR; term *fault management* is used in MND-TMM; *defect management* is present in TPI, TPI NEXT, TPI Automotive and ATG add-on for TPI, while the term *fault/defect handling* is found in TAIM.

Many test models emphasize **environment** as one of their process areas. MB-VV-MM, TMMi, MND-TMM, TPI, TPI NEXT, TPI Automotive model, ATG add-on for TPI model, Test SPICE, Software testing standards ISO/IEC 29119 and ISO 33063 and TAIM use the term *test environment*, while TPI, TPI Automotive and ATG add-on for TPI also state *office environment* as process area.

Test design is emphasized by TMMi, MB-VV-MM, TPI NEXT, TPI Automotive, Software testing standards ISO/IEC 29119 and ISO 33063, ATP, ISTQB test process model and TAIM and in CenPRA, as a part of test preparation. **Test goals** are important areas in: TAIM, MB-VV-MM and TMM; **policy** is found in TMM, TMMi, MB-VV-MM, Software testing standards ISO/IEC 29119 and ISO 33063, while **strategy** process area is present in TMMi, MND-TMM, TPI, TPI NEXT, TPI Automotive and ATG add-on for TPI, Software testing standards ISO/IEC 29119 and ISO 33063.

Therefore, the answer to RQ1 (*What common process areas can be identified and how are they presented in different models?*) is that test models have common process areas like test planning, quality, defect management, environment, test design, test goals, policies, strategies. However, some test process areas are presented differently; some models require "deeper" insight e.g. V model and CenPRA, where test planning is in the area of test preparation. Different test models use synonyms for defect management and software quality evaluation; some have quality assurance, quality management or test quality.

Regarding RQ2 (*Are there inconsistencies among maturity levels and how significant are they?*), the answer is yes. There are inconsistencies among maturity levels of different test models. Firstly, they do not agree on definition of starting point; e.g. TMM's initial level means ad hoc, chaotic testing processes, while MPT.BR assumes that testing is partially managed on the initial level. Secondly, meaning of "optimized" differs, which means that e.g. if an organization reaches level 5 of MPT.BR, it will be only on the level 4 of CMMi and Afzal et al. (2016) found that if an organization reaches the highest maturity level of TPI NEXT, it will not surpass level 2 of TMMi.

When it comes to practitioner-test model relation, the answer to RQ3 (*Why are practitioners reluctant in accepting test process models?*) is that a lot of test process models are available, but are either general or domain specific, or are just on a conceptual level and simply cannot be applied in practice (Afzal et al., 2016). Also, different models vary in maturity levels and process areas. These variations become a source of confusion. Therefore, practitioners conclude that it is the easiest to just create a new model, regardless of possibility that a similar model already exists, and that the resources invested in creating it were unnecessary.

Answer to RQ4 (*How can model confusion be minimized or eliminated?*) is that, due to variations in maturity levels and process areas, confusion can be minimized by careful, detailed examination of test process models, since many models have common process areas (e.g. test planning). As far as maturity levels are concerned, each software organization should choose one that best fits its needs and stick with it. Otherwise, it will not be able to monitor its test process

improvement progress and take necessary actions for further improvement, since reaching the highest maturity level in one model, does not necessarily means the same when current test practice is compared to another maturity model.
In industry-test model relation, the answer to RQ5 (*Can identified common test process areas and maturity level variations help in building more industry appropriate test process models?*) is yes, by incorporating common process areas and standardizing meanings of maturity levels in "new" test models, which also have to take into consideration the type of industry they are being developed for. These factors will eliminate confusion, unwillingness and rejection of practitioners towards standardization of testing process.

6. CONCLUSION

A lot of test process models have been developed with the purpose of improving test process quality and thus, software quality. Ironically, instead uniformity, these models brought variations in maturity levels and process areas and thus, a vast gap between academia and industry. Organizations are confused regarding where they are in terms of improvement of testing processes and what actions should be taken to reach the best practice in the software development industry. Therefore, the purpose of this research was to extract and interpret maturity level variations and identify common process areas. It was noted that there are inconsistencies regarding starting points in maturity levels; that reaching a higher maturity level in one model, will still mean staying on the lower level of another model and that level "optimization" has different meanings in different models. Test process models have in common process areas like test planning, quality, defect management and etc., but confusion can arise if models are not thoroughly examined do to the scope of process areas and used synonyms. All the above stated just confirms that no one model fits all test process improvement needs and that there can be many reasons for such state: some models from academia are not based on real, industrial needs, but "hypothetically argued motivations" or because best practices from academia and industry are not taken into account as Garousi et al., stated in 2017 and 2018. Therefore, the only action that software companies can take is to choose one model that fits them best and monitor their test process improvement accordingly.

REFERENCES

(2007). The pain and the gain. In M. Utting & B. Legeard (Eds.), Practical Model-Based Testing: A Tools Approach (pp. 19-57).

Afzal, W., Alone, S., Glocksien, K., & Torkar, R. (2016). Software test process improvement approaches: A systematic literature review and an industrial case study. The Journal of Systems and Software, 111, 1-33.

Bueno, P.M.S., Crespo, A.N., & Jino, M. (2006). Analysis of an Artifact Oriented Test Process Model and of Testing Aspects of CMMI. 7th International Conference on Product Focused Software Process Improvement 2006 (pages. 263-277). Amsterdam

Burnstein, I., Suwannasart, T., & Carlson, C.R. (1996). Developing a Testing Maturity Model, Part II. Retrieved April 10, 2020, from https://pdfs.semanticscholar.org/db39/93ef31a0112f45d617b4ca412ce8ebe297ef.pdf

Eldh, S., Andersson, K., & Wiklund, K. (2014). Towards a Test Automation Improvement Model (TAIM). IEEE Seventh International Conference on Software Testing, Verification, and Validation Workshops 2014 (pages. 337-342). Cleveland: IEEE.

Ericson, T., Subotic, A., & Ursing, S. (1997). TIM - A Test Improvement Model. Software Testing, Verification and Reliability 7 (4), 229–246.

Furtado, A.P.C.C., Gomes, M.A.W., Andrade, E.C., & de Farias, I.H. (2012). MPT. BR: A Brazilian Maturity Model for Testing. 12th International Conference on Quality Software 2012 (pages. 220–229). Xi'an, Shaanxi: IEEE.

Garcia, C., Dávila, A., & Pessoa, M. (2014). Test Process Models: Systematic Literature Review. 14th International Conference on Software Process Improvement and Capability Determination 2014 (pages. 84–93). Vilnius.

Garousi, V., & Küçük, B. (2018). Smells in software test code: A survey of knowledge in industry and academia. The Journal of Systems and Software, 138, 52-81.

Garousi, V., Felderer, M., & Hacaloglu, T. (2017). Software test maturity assessment and test process improvement: A multivocal literature review. Information and Software Technology, 85, 16-42.

Garousi, V., Felderer, M., & Hacaloglu, T. (2018). What We Know about Software Test Maturity and Test Process Improvement. IEEE Software, 35 (1), 84-92.

Garousi, V., Felderer, M., Karapıcak, C. M., & Yılmaz, U. (2018). Testing embedded software: a survey of the literature. Information and Software Technology, 104, 14-45.

Heiskanen, H., Maunumaa, M., & Katara, M. (2012). A Test Process Improvement Model for Automated Test Generation. 13th International Conference on Product-Focused Software Process Improvement 2012 (pages. 17–31). Madrid: Springer.

Imtiaz, J., Sherin, S., Khan, M. U., & Iqbal, M. Z. (2019). A systematic literature review of test breakage prevention and repair techniques. Information and Software Technology, 113, 1-19.

Jung, E. (2009). A Test Process Improvement Model for Embedded Software Developments. Ninth International Conference on Quality Software 2009 (pages. 432-437). Jeju: IEEE.

Karlström, D., Runeson, P., & Nordén, S. (2005). A minimal test practice framework for emerging software organizations. Journal of Software Testing, Verification and Reliability 15 (3), 145–166.

Kasurinen, J. (2012). Software Organizations and Test Process Development. Advances in Computers, 85, 1-63.

Kasurinen, J., Runeson, P., Riungu, L., & Smolander, K. (2011). A Self-assessment Framework for Finding Improvement Objectives with ISO/IEC 29119 Test Standard. European Conference on Software Process Improvement 2011 (pages. 25-36). Roskilde: Springer.

Oh, H., Choi, B., Han, H., & Wong, W. E. (2008). Optimizing Test Process Action Plans by Blending Testing Maturity Model and Design of Experiments. The Eighth International Conference on Quality Software 2008 (pages. 57-66). Oxford: IEEE.

Ryu, H., Ryu, D., & Baik, J. (2008). A Strategic Test Process Improvement Approach Using an Ontological Description for MND-TMM. Seventh IEEE/ACIS International Conference on Computer and Information Science 2008 (pages. 561-566). Portland: IEEE.

Sánchez-Gordón, ML., & Colomo-Palacios, R. (2018). From Certifications to International Standards in Software Testing: Mapping from ISQTB to ISO/IEC/IEEE 29119-2. 25th European Conference on Systems, Software and Services Process Improvement 2018 (pages. 43-55). Bilbao: Springer.

Steiner, M., Blaschke, M., Philipp, M., & Schweigert, T. (2012). Make test process assessment similar to software process assessment - the Test SPICE approach. Journal of Software: Evolution and Process, 24, 471-480.

van Veenendaal, E., & Cannegieter, J.J. (n.d.). Test Maturity Model integration (TMMi) Results of the first TMMi benchmark – where are we today?. Retrieved April 10, 2020, from https://www.improveqs.nl/media/1144/e-book_tmmi-2013.pdf

von Mayrhauser, A., & Mraz, R. (1995). The Sleuth approach to aerospace software testing. IEEE Aerospace Applications Conference 1995 (pages. 61-75). Aspen, USA: IEEE.

von Mayrhauser, A., Walls, J., & Mraz, R. (1994). Testing applications using domain based testing and Sleuth. IEEE International Symposium on Software Reliability Engineering 1994 (pages. 206-215). Monterey: IEEE.

Yang, J., Yuan, Y., & Zhang, T. (2015). Research on Modeling of Software Automatic Test. International Conference on Trustworthy Computing and Services 2014 (pages. 382-388). Beijing: Springer.