Camargo *et al. Journal of Software Engineering Research and Development* (2015) 3:7
DOI 10.1186/s40411-015-0019-9

Journal of Software Engineering Research and Development
a SpringerOpen Journal

RESEARCH Open Access

Characterising the state of the practice in software testing through a TMMi-based process

Kamilla G Camargo*, Fabiano C Ferrari and Sandra CPF Fabbri

*Correspondence:
kcamargo@gmail.com
Computing Department, Federal University of São Carlos, Rod. Washington Luis, km 235, 13565-905 Sao Carlos, SP, Brazil

Abstract

Background: The software testing phase, despite its importance, is usually compromised by the lack of planning and resources in industry. This can risk the quality of the derived products. The identification of mandatory testing-related practices may lead to the definition of feasible processes for varied sizes of software companies. In this context, this work aims at identifying a set of key practices to support the definition of a generic, streamlined software testing process, based on practices that are described in the TMMi (Test Maturity Model integration), and verify the alignment of the devised process with the TMMi levels.

Methods: We have performed 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 process; additionally, we applied a diagnosis tool in order to measure the level of TMMi which is fulfilled with the devised process.

Results: A set of 33 (out of 81) practices were ranked as mandatory by most of participants, which represents 40 % of the TMMi's full set of practices; on the downside, a testing process that relies on this subset of TMMi practices does not fully fulfil level 2 (Managed) of the maturity model.

Conclusions: The identified subset of practices can guide the definition of a lean testing process when compared to a process that includes all TMMi practices; it is expected that such a process encourages a wider adoption of testing activities in software development; even though the streamlined process does not encompass many practices that are spread across TMMi levels, a substantial subset of level 2 practices (Managed) should be accomplished with its adoption.

Keywords: Software testing process; TMMi practices

1 Introduction

Since software had become widely used, it has played an important role in the people's daily lives. Consequently, its reliability cannot be ignored (Cao et al. 2010). In this context, quality assurance (QA) activities should monitor the whole development process, promoting the improvement of the final product quality, and hence making it more reliable. One of the main QA activities is software testing that, when well executed, may deliver a final product with a low number of defects.

Despite the importance of software testing, many companies face difficulties in devising a software testing process and customising it to their reality. A major barrier is the difficulty in adapting testing maturity models for the specific environment of an organisation (Rodrigues et al. 2010). Many organisations recognise that process improvement initiatives can solve these problems. However, in practice, defining the steps that can be taken to improve and control the testing process phases and the order they should be implemented is, in general, a difficult task (Andersin 2004).

Reference models, such as TMMi (TMMI Foudation 2012) and MPT.Br (Softex Recife 2011), point out what should be done for the improvement of the software testing process. Despite the organisation of the models in levels (such as in CMMI (SEI 2006)), which suggests an incremental implementation from the lowest level, reference models for testing process improvement have a large number of practices that must be satisfied, though not all of them are feasible for all sizes of companies and teams. In addition, the establishment of a testing process relying on a reference model becomes a hard task due to the difficulty for model comprehension. Moreover, the models do not define priorities in case of lack of time and/or resources, thus hindering the whole model adoption.

According to Purper (2000), the team responsible to define the testing process usually outlines a *mind map* of the model requirements in relation to the desired testing process. This team manually verifies whether mandatory practices required by the model are addressed. In general, these models indicate some prioritisation through their levels; however, within each level, it is not clear what should be satisfied at first.

For better results, the testing process should include all phases of software testing. However, the process should be as minimal as possible, according to the reality of the company and the model used for software development. This adequacy can make the testing process easier to be applied and does not require many resources or a large team. This should help the company achieve the goal of improving the product quality.

Based on this scenario, we conducted a survey in order to identify which are the practices of TMMi that should be always present in a testing process. Our goal was to characterise the context of a sample of Brazilian companies to provide them with a direction on how to define a lightweight, still complete testing process. Therefore, the survey results reflect the point of view of the analysed group of Brazilian testing professionals. Given that a generic testing process encompasses phases such as planning, test case design, execution and analysis, and monitoring (Crespo et al. 2010; Hass 2008), we expected the survey could indicate which are the essential practices for each phase. The assumption was that there are basic activities related to each of these phases that should never be put aside, even though budget, time or staff are scarce. We upfront highlight that our choice for TMMi as the baseline reference model was mainly motivated by its worldwide scope and multi-context adoption (Experimentus Ltd. 2012); furthermore, it is well acknowledged that more specific models such as MPT.Br are usually inspired in widely adopted models such as CMMI and TMMi (Softex Recife 2011).

This paper extends the results originally presented in our previous paper (Camargo et al. 2013). In order to extend our previous work, we performed an extra analysis. In this new analysis, our aim was to verify which TMMi level a software company that runs the proposed streamlined process may achieve in case such company is pursuing a TMMi certification. We used a testing process diagnosis tool called KITTool (Höhn 2011) to support the data collection. KITTool allows one to visualise which TMMi practices and levels are fulfilled in a given context.

The remainder of this paper is organised as follows: Section 2 describes the concepts that underlie this research. Section 3 presents the survey planning, the process we adopted to invite participants to answer the survey, and the methods applied to evaluate the gathered data. Section 4 shows the survey results and the participant's profile characterisation. Section 4.3 analyses these results for each phase of the generic, streamlined testing process. In the sequence, Section 5 analyses the results from the perspective of the TMMi maturity levels. Finally, Section 6 presents possible threats to the validity of the survey, and Section 7 presents the conclusions.

2 Background

TMMi (TMMI Foudation 2012) is a reference model that complements CMMI (SEI 2006) and was established to guide the implementation and improvement of testing processes. It is similar to CMMI in structure, because it includes maturity levels that are reached through the achievement of goals and practices. For TMMi, a process evolves from a chaotic initial state (Level 1), to a state in which the process is managed, controlled and optimised (Level 5). Each specific goal indicates a single characteristic that must be present in order to satisfy the corresponding process area. A specific goal is divided into specific practices that describe which activities are important and can be performed to achieve the goal. Generic goals are related to more than one process area and describe features which may be used to institutionalise the testing process. Figure 1 illustrates the structure of TMMi. The survey questionnaire of this study was developed based on the TMMi specific goals and practices. Each goal was represented by a question and each practice represented by a sub question.

Höhn (2011) defined a mind map of TMMi. The map distributes process areas, specific goals and their practices throughout phases of a generic testing process. This map is called KITMap and was developed to facilitate the TMMi understanding and to share information. This work applied KITMap tool for fist time in industry. In the map, the root node is the name of the treated theme, *i.e.* the testing process. Nodes of the second level are the phases of a generic testing process. Such phases guided the grouping of the survey questions. They are: Planning, Test Case Design, Setup of Test Environment and Data, Execution and Evaluation, and Monitoring and Control. At the third level of KITMap are process areas of TMMi.

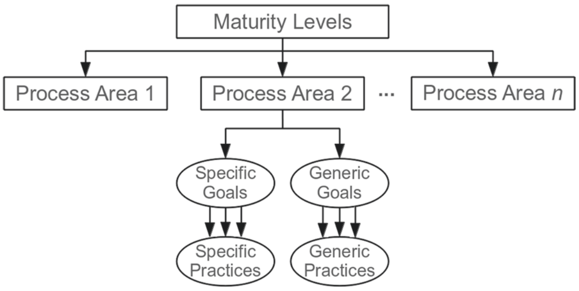


Fig. 1 TMMi structure and components (TMMI Foudation 2012)

Höhn (2011) organised the process areas of TMMi according to their relation to each phase of the generic testing process. Figure 2 illustrates, from the left side, (i) the phase of the generic testing process (*i.e.* Test Case Design); (ii) two process areas that are related to that phase; (iii) the specific goal related to the first process area; and (iv) the various specific practices related to the specific goal. Note that process areas from different TMMi levels may be associated to the same phase of the generic testing process. This can be observed in Fig. 2, in which one process area is from Level 2 of TMMi while the other is from Level 3. Despite this, both are associated to the same phase (Test Case Design).

3 Methods

This research was performed with the aim of identifying which are the most important practices of TMMi, and hence should be prioritised during the testing process execution, according to the opinion of experienced testing professionals. This study is motivated by our experience and, equally important, real life observations that some testing-related practices should never be put aside, even though time, budget or human resources are scarce. In addition, the intended, narrowed testing process can be used as a starting point for the implementation of TMMi in a company who intends to obtain the certification. Even though the process might not fulfil a given maturity level of the model, it should include the most relevant practices and can be further improved to climb TMMi levels.

3.1 Survey design

The survey was developed using the *Lime Survey* tool (LimeSurvey 2014). Lime Survey allows one to organise survey questions in groups and visualise them in separate web pages. The questionnaire is based on TMMi Version 3.1 (TMMI Foudation 2012).

Questions were split into six groups. The first group aims to characterise subject profiles; it includes questions related to the level of knowledge on software quality reference models, namely, CMMI (SEI 2006), MR-MPS (Softex 2012) and TMMi. The remaining groups of questions (*i.e.* 2 to 6) each focuses on a phase of a generic testing process, as defined by Höhn (2011). The phases are: (1) Planning (2) Test Case Design (3) Setup of Testing Environment and Data (4) Execution and Evaluation (5) Monitoring and Control.

Each questionnaire page includes a single group of questions. The first page also brings some directions regarding how to fill in the forms, including a table to describe the values the subjects could use to assign each TMMi practice a level of importance. The values are described in Table 1. Note that we decided not to include a neutral value for the scale of importance. This was intended to make the subject decide between practices that should be classified as priority (i.e. levels 4 or 3 of importance) or not (i.e. levels 2 or 1). All questions related to testing practices (*i.e.* groups 2 to 6) are required; otherwise, a subject cannot go ahead to the next group of questions.

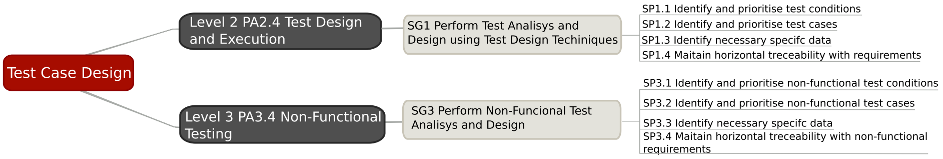

Test Case Design
Level 2 PA2.4 Test Design and Execution
SG1 Perform Test Analisys and Design using Test Design Techiniques
SP1.1 Identify and prioritise test conditions
SP1.2 Identify and prioritise test cases
SP1.3 Identify necessary specifc data
SP1.4 Maitain horizontal treceability with requirements
Level 3 PA3.4 Non-Functional Testing
SG3 Perform Non-Funcional Test Analisys and Design
SP3.1 Identify and prioritise non-functional test conditions
SP3.2 Identify and prioritise non-functional test cases
SP3.3 Identify necessary specifc data
SP3.4 Maitain horizontal treceability with non-functional requirements

Fig. 2 KITMap excerpt (adapted from the work of Höhn (2011))

Table 1 Levels of importance for surveyed practices

1- Dispensable	Dispensable activity; does not need to be performed.
2- Optional	Activity that not necessarily needs to be performed.
3- Desirable	Activity that should be implemented, though may be put aside.
4- Mandatory	Essential activity that must always be performed.

We highlight two key points in this survey: (1) *none of the subjects were told the questionnaire was based on the TMMi structure* – this intended to avoid bias introduced by knowledge on the process maturity model; and (2) *the subjects should answer the questionnaire according to their personal opinion* – this intended to avoid bias introduced by the company or institution context.

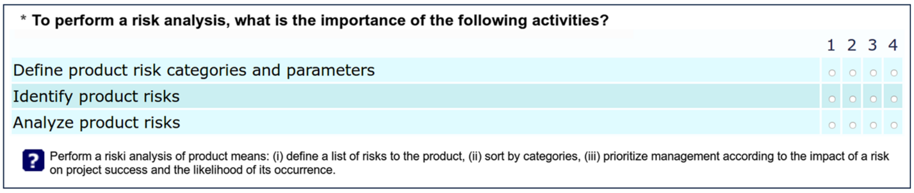
To build the questionnaire, which is available online (Camargo 2012), we translated TMMi goals, practices and other items of interest to Portuguese, since there is no official translation of TMMi to languages other than English. The translation took into account technical vocabulary in the target language (*i.e.* Portuguese).

In the questionnaire, every question within a given group (*i.e.* within a specific testing process phase) regards a TMMi *Specific Goal*. Each question includes a set of subquestions regarding the TMMi *Specific Practices* (SPs). Note that in TMMi a Specific Goal is achieved when the associated SPs are performed in a testing process. Therefore, assigning a particular set of SPs levels of importance should allow us to also draw conclusions about the Specific Goal relevance, according to the subject's personal experience.

Figure 3 illustrates a question related to the Planning Phase. This question addresses the *Perform a Product Risk Assessment* Specific Goal, and includes three sub-questions regarding the associated SP's. As previously described, the subject should assign a level of importance ranging from 1 to 4 to each SP (see Table 1), according to its opinion about the relevance of the SP to achieving the goal defined in the question. Note that all questions bring a side note to help the subject understand and properly answer the question. This help note can be seen in the bottom of Fig. 3.

Characterising profiles: The first group of questions aims to characterise the profile of subjects taking into account their work environment. Figure 4 shows part of the profile form. To design the profile form, we considered that the subject's experience and the process maturity level of its institution or company impact on the subject's knowledge on testing. Therefore, the following information is required:

- Experience with software testing (research, industry and teaching): it is well-known that tacit knowledge is different from explicit knowledge. Due to this, this

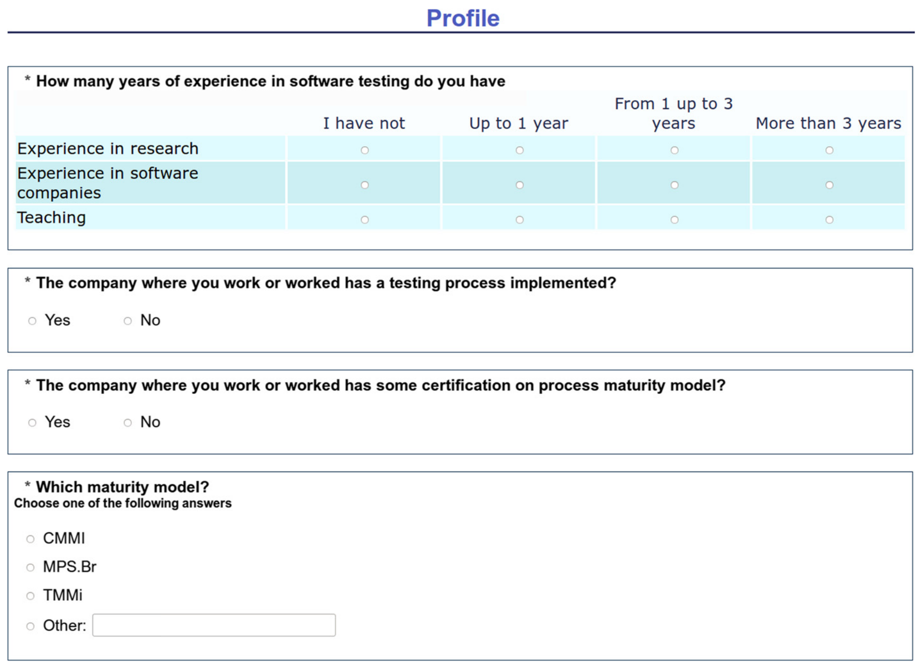


* **To perform a risk analysis, what is the importance of the following activities?**

	1	2	3	4
Define product risk categories and parameters	○	○	○	○
Identify product risks	○	○	○	○
Analyze product risks	○	○	○	○

? Perform a riski analysis of product means: (i) define a list of risks to the product, (ii) sort by categories, (iii) prioritize management according to the impact of a risk on project success and the likelihood of its occurrence.

Fig. 3 Example of question, structured according to a testing process phase and TMMi Specific Goals and Practices



Profile

* **How many years of experience in software testing do you have**

	I have not	Up to 1 year	From 1 up to 3 years	More than 3 years
Experience in research	○	○	○	○
Experience in software companies	○	○	○	○
Teaching	○	○	○	○

* **The company where you work or worked has a testing process implemented?**

○ Yes ○ No

* **The company where you work or worked has some certification on process maturity model?**

○ Yes ○ No

* **Which maturity model?**
Choose one of the following answers

○ CMMI
○ MPS.Br
○ TMMi
○ Other: □

Fig. 4 Part of the profile characterisation form (translated to English)

information aims to characterise different types of knowledge, acquired either with industrial, research or teaching experience.

- Testing process in the company: this information is required only for those who report experience in industry, in order to characterise their work environment.
- Certification in process maturity model: this information is required for those who report their companies have any certification in maturity models; if applicable, the subject is required to inform which maturity model (namely, MR-MPS, CMMI, TMMi or any other) and the corresponding level. This might have impact on the subject's personal maturity regarding the model.
- Knowledge of TMMi and MR-MPS: knowledge of reference models, specially TMMi, grants the subject a higher maturity regarding testing processes.

3.2 Obtained sample

For this survey, a personal e-mail announcement was sent to Brazilian software testing professionals from both academy and industry. It was also announced in a mailing list (Melo 2004) that includes more than 3,000 subscribers from Brazil. Furthermore, we invited professionals that work for a pool of IT companies named PISO (Pólo Industrial de Software) (PISO 2015) from the city of Ribeirão Preto, Brazil.

The questionnaire was made available in December, 2011, and remained open for a period of 45 days. In total, we registered 113 visits, from which 39 resulted in fully answered questionnaires that were considered for data analysis. Even though the sample is not large, these 39 answers allowed us to analyse the data statistically, however with less rigour than in analyses applied to large samples. The analysis procedures are described in the following section.

3.3 Data analysis method

Initial analysis: An initial data analysis revealed that practices were mostly ranked as 3 and 4 in regard to their level of importance. This is depicted in Fig. 5, which groups the answers of all subjects for all questions according to the assigned levels of importance[1]. This initial analysis also allowed us to identify two outliers which were removed from the dataset: the first regards a subject that assigned level 4 to all practices, while the second inverted all values in his/her answers (*i.e.* he/she interpreted the value of 4 as the lowest level of importance and the value of 1 as the highest level). Therefore, the final dataset, depicted in Fig. 5, comprises 37 fully filled in questionnaires.

In this survey, we considered the following independent variables: (i) industrial experience with testing process; (ii) knowledge and usage experience with MR-MPS; and (iii) knowledge of TMMi. The dependent variable is the level of importance assigned to each practice. The scale used for the dependent variable characterises data with ordinal measurement level, *i.e.* we were dealing with discrete values. Besides, the data distribution was non-symmetric since the vast majority of practices were ranked as 3 and 4, as shown in Fig. 5.

The characteristics of the data led us to use the non-parametric *Sign Test* (Whitley and Ball 2002). This test evaluates if the median, for a given set of values (in our case, *for each practice*) is higher than a fixed value. We used the fixed value of 3.5, since the overall median was approximately 3. Note that the fixed value is higher than the overall median, so the Sign Test would allow us to identify each practice whose median ranking was closer to 4 (*i.e.* mostly ranked as mandatory), since more than 50 % of subjects would have been ranked those practices with maximum level of importance.

Due to the size of our sample, we adopted a *p-value*=0.15 to draw conclusions on the executed tests. Even though this is not a widely adopted level of confidence, some other exploratory studies (Basili and Reiter 1981; Miller 2004), which dealt with similar small samples, also adopted relaxed levels of confidence instead of the traditional *p-value*=0.01 or *p-value*=0.05.

The results of this analysis did not result in statistical significance for some practices, even when the majority of subjects assigned levels 3 or 4 for those practices. For instance, the ***Identify and prioritise test cases*** practice was ranked as mandatory by most of subjects (19 out of 37); however, the Sign Test did not show statistical significance. Obviously, the sample size may have impacted on the sensitiveness of the statistical test, leading to

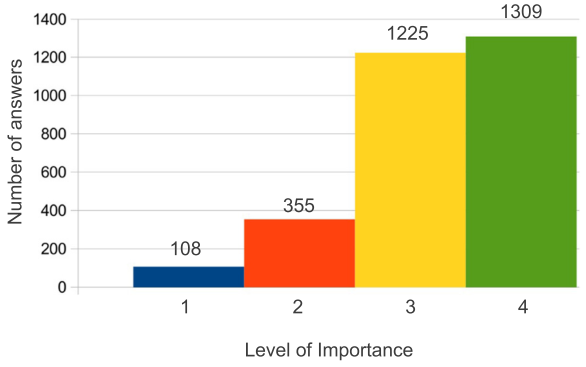


Fig. 5 Frequency distribution of levels of importance, considering all answers of all subjects

inconclusive results even in cases of majority of answers ranging from 3 to 4. This is the case of the ***Identify and prioritise test conditions*** practice. The answer distribution for this practice is summarised in Table 2. The figures show that the number of subjects that assigned this practice level 3 of importance is higher than the number of subjects that assigned it level 4; despite this, we could not observe any difference, statistically significant, in favour of the former (*i.e.* level 3).

A new analysis, based on the frequency of answers in the first set of questions, indicated some trends the statistical tests did not allow for. It consisted in a descriptive analysis of the data, since we were unable to conclude on some practices, even when they were ranked as mandatory by many subjects. In short, we identified the practices that were mostly ranked as mandatory when comparing to the other values of the scale (desirable, optional and dispensable – see Table 1). In spite of the weak confidence such kind of analysis may represent, the identified subset of practices was similar to the subset obtained solely based on a statistical basis. In fact, this set of practices included all practices identified through the statistical aforementioned procedures. A summary of results is depicted in the Venn diagram of Fig. 7. Details are discussed in Sections 4 and 4.3.

Second questionnaire: After these aforementioned analyses, we elaborated a new set of three questions to help us clarify some open issues. These new questions, which require simple "Yes" or "No" answers, aimed to resolve some dependencies observed in the results. Such dependencies were identified by Höhn (2011) and indicate that the implementation of some testing-related practices require the previous implementation of others.

The open issues were related with the following practices: ***Analyse product risks***, ***Define the test approach*** and ***Define exit criteria***. The motivation for this new questionnaire and the analysis of results are presented in Section 4.3.

This new questionnaire was announced by e-mail to all subjects who answered the first one, and remained open for a period of 14 days. We had feedback from 14 subjects (see Section 4.3).

Additional analysis: In order to analyse and discuss the TMMi level that can be achieved with the devised streamlined testing process, we used a tool called KITTool (Knowledge and Improvement on Test - Tool) (Höhn 2011). KITTool is able to show the fulfilment degree for each TMMi level according to the set of specific practices currently implemented, either fully or partially. The tool allows testing analysts or engineers to analyse the current testing process based on TMMi's specific practices or specific goals. Beyond that, the tool points out the path for a new process definition based on the obtained diagnosis.

Table 2 Levels of importance assigned to the *Identify and prioritise test conditions* practice

Level	Number of answers
4	14
3	19
2	2
1	2

Using KITTool, an analyst or engineer can assign each practice a grade that ranges from 0 to 100 %, according to its focus and application status within the testing process (Höhn 2011). In short, if a given practice is partially fulfilled for some – but not all – projects, and there is no clear procedure on how to execute it, its assigned a grade of 25 %. Practices are assigned a grade of 50 % when there are a defined procedure for them, however they are not fully documented and are only applied to strategic projects of a company. If a practice is formally defined in terms of procedures and documentation, and that practice is applied to the majority of projects (*e.g.* more than 80 %), its assigned a grade of 75 %. To conclude, 100 % is assigned to practices that are fully defined and applied in all projects.

To perform the new analysis presented in this paper, as an extension of our previous work (Camargo et al. 2013), we used KITTool to assign every practice in the obtained streamlined testing process a grade of 100 %. That is, we consider the case of a hypothetical software company that fully runs the streamlined process, for all projects, having adequate documentation and well-defined procedures for all practices. With such diagnosis at hand, we analyse how close (or far) such a company is to achieve a given TMMi maturity level. Note that, for this analysis, we filled in KITTool to grade all practices of the streamlined process as 100 % accomplished. Furthermore, note that this analyses was done after the original publication of the survey results (Camargo et al. 2013).

4 Results

The results of the survey are described in this section. Initially, Section 4.1 defines some profiles, each representing a group of subjects, based on the experience reported in the profile characterisation form. Then, Section 4.2 shows the results with respect to the level of importance of TMMi practices according to each profile.

4.1 Profile definition

Figure 6 summarises the level of knowledge of both subjects and their institutions according to the profile characterisation questions. A description of the charts comes in the sequence.

a) **Experience**: this chart shows that 46 % of subjects (17 out of 37) have more than three years of experience in testing either in industry or academy; only 11 % (4 out of 37) have less than one-year experience.

b) **Testing Process**: this chart shows that 65 % of subjects (24 out of 37) work (or have worked) in a company that has a testing process officially implemented (*i.e.* an explicit testing process). From the remaining subjects, 22 % (8 out of 37) do not (or have not) worked in a company with an explicit testing process, while around 14 % (5 out of 37) have not answered this question.

c) **Certification**: this chart shows that 59 % of subjects (22 out of 37) work (or have worked) in a company that has been certified with respect to a software process maturity model (*e.g.* CMMI, MR-MPS). The remaining subjects have never worked in a certified company (24 %) or have not answered this question (16 %).

d) **Type of Certification**: from the subjects that reported to work (or have worked) in a certified company – chart (c) of Fig. 6 –, half of them (*i.e.* 11 subjects) are (or were) in a CMMI-certified company, while the remaining are (or were) in a MR-MPS-certified company.

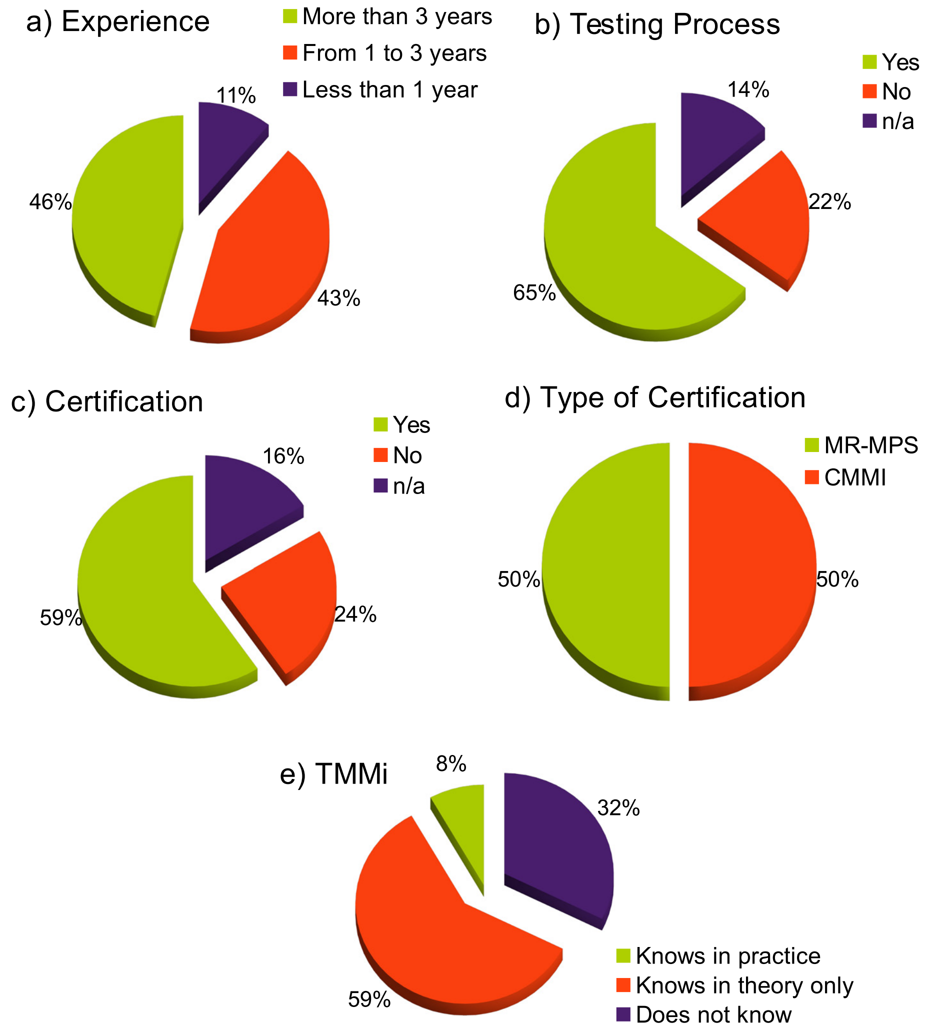


Fig. 6 Summary of profile characterisation

e) **TMMi**: this chart reveals that only 8 % of subjects (3 out of 37) have had any practical experience with TMMi. Besides this, 59 % of subjects (22 out of 37) have stated to have only theoretical knowledge of TMMi, whereas 32 % (12 out of 37) do not know this reference model.

Based on the results depicted in Fig. 6, we concluded that the sample is relevant with respect to the goals established for this work. This conclusion relies on the fact that, amongst the 37 subjects who have fully answered the questionnaire, (i) 89 % of them have good to high knowledge of software testing (*i.e.* more than one-year experience); (ii) 65 % work (or have worked) in companies that officially have a software testing process; (iii) 59 % work (or have worked) in a CMMI- or MR-MPS-certified company; and (iiii) 67 % are knowledgeable of TMMi, at least in theory. For CMMI-certified companies, the maturity levels vary from 2 to 5 (*i.e.* from *Managed* to *Optimising*). For MR-MPS-certified companies, the maturity levels range from G to E (*i.e.* from *Partially Managed* to *Partially Defined*).

To analyse the results regarding the level of importance of TMMi practices according to the subjects' personal opinion, we defined three different profiles as follows:

- Profile-Specialist: compound by 12 subjects who have at least three years of experience with software testing and work (or have worked) in a company that has a formally implemented software testing process.
- Profile-MR-MPS: compound by 20 subjects that are knowledgeable of MR-MPS and use this reference model in practice.
- Profile-TMMi: compound by 25 subjects that are knowledgeable of TMMi.

The choice for a MPS.BR-related profile was motivated by the straight relationship between the reference model and context of Brazilian software companies. Furthermore, these three specific profiles were defined because we believe the associated subjects' tacit knowledge is very representative. Note that the opinion of experts in CMMI was not overlooked at all; instead, such experts' opinion are spread over the analysed profiles. Finally, we also considered the answers of all subjects, in a group named Complete Set.

4.2 Characterising the importance of TMMi practices

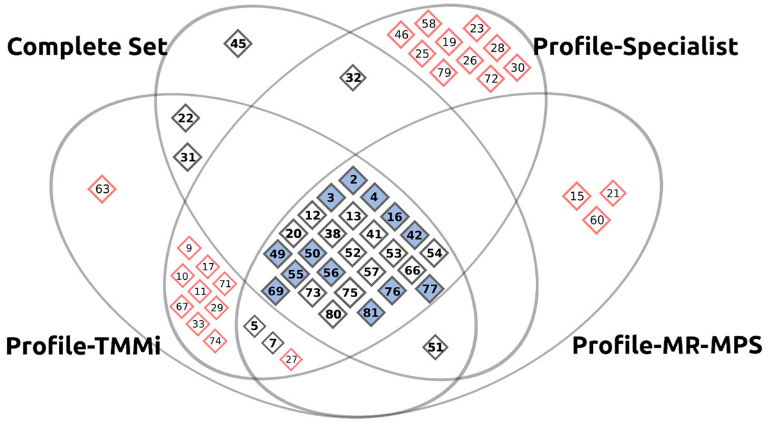
As previously mentioned, the results herein described are based on the three profiles (namely, Profile-Specialist, Profile-MR-MPS and Profile-TMMi) as well as on the whole survey sample. Within each profile, we identified which practices were mostly ranked as mandatory. The Venn diagram depicted in Fig. 7 includes all mandatory practices, according to each profile. The practices are represented by numbers and are listed in the table shown together with the diagram.

In Fig. 7, the practices with grey background are also present in the set obtained solely from the statistical analysis described in Section 3.3. As the reader can notice, this set of practices appears in the intersection of all profiles. Furthermore, practices with bold labels (*e.g.* practices 5, 7, 22, 31 etc.) are present in the set aimed to compose a lean testing process (this is analysed in details in Section 4.3). Next we describe the results depicted in Fig. 7.

- Complete Set: taking the full sample into account, 31 practices were assigned level 4 of importance (*i.e.* ranked as mandatory) by most of the subjects. The majority of them are also present in the other profile-specific sets, as shown in Fig. 7. The reduced set of practices to compose a lean testing process includes these 31 items, and is complemented with practices #5 and #7 (the justification is presented in Section 4.3).
- Profile-Specialist: 49 practices were ranked as mandatory by most of subjects within this profile. From these, 27 practices appear in the intersection with at least another set;
- Profile-MR-MPS: subjects of this profile ranked 33 practices as mandatory, from which only 30 are in intersections with the other profiles; only 3 practices are considered mandatory exclusively for subjects of this profile.
- Profile-TMMi: for those who know TMMi, 42 practices are mandatory, from which 41 ones appear in the intersections with the other profiles.

4.3 The obtained streamlined process

Before the definition of the aimed reduced set of practices, we analysed the results of the second questionnaire, which has been designed to resolve some dependencies observed



2	Identify product risks	42	Implement the test environment
3	Analyse product risks	45	Perform test environment intake test
4	Identify items and features to be tested	46	Develop and prioritise non-functional test procedures
5	Define the test approach	49	Execute test cases
7	Define exit criteria	50	Report test incidents
9	Establish a top-level work breakdown structure	51	Write test log
10	Define test lifecycle	52	Decide disposition of test incidents in configuration control board
11	Determine estimates for test effort and cost	53	Perform appropriate action close the test incident
12	Establish the test schedule	54	Track the status of test incidents
13	Plan for testing staffing	55	Execute non-functional test cases
15	Identify test project risks	56	Report non-functional test incidents
16	Establish the test plan	57	Write test log (non-functional)
17	Review test plan	58	Conduct peer reviews
19	Obtain test plan commitments	60	Analyse peer review data
20	Elicit test environment needs	63	Monitor test commitments
21	Develop the test environment requirements	66	Conduct test progress reviews
22	Analyse the test environment requirements	67	Conduct test progress milestone reviews
23	Identify non-functional product risks	69	Monitor defects
25	Identify non-functional features to be tested	71	Monitor exit criteria
26	Define the non-functional test approach	72	Monitor suspension and resumption criteria
27	Define non-functional exit criteria	73	Conduct product quality reviews
28	Identify work products to be reviewed	74	Conduct product quality milestone reviews
29	Define peer review criteria	75	Analyse issues
30	Identify and prioritise test conditions	76	Take corrective action
31	Identify and prioritise test cases	77	Manage corrective action
32	Identify necessary specific test data	79	Perform test data management
33	Maintain horizontal traceability with requirements	80	Co-ordinate the availability and usage of the test environments
38	Develop and prioritise test procedures	81	Report and manage test environment incidents
41	Develop test execution schedule		

Fig. 7 Venn diagram – intersections of results within each profile

in the initial dataset (*i.e.* based on the 37 analysed answers). The dependencies have been identified by Höhn (2011), who pointed out some practices that must be implemented before the implementation of others. Based on the feedback of 14 subjects, all included in the initial sample, we were able to resolve the observed dependencies, which are related to the following practices: ***Analyse product risks***, ***Define the test approach***, and ***Define exit criteria***.

Regarding ***Analyse product risks*** (practice #3 in Fig. 7), the subjects were asked if this task should be done as part of the testing process. We got 12 positive answers, thus indicating this practice is relevant, for example, to support the prioritisation of test cases. In fact, the ***Analyse product risks*** practice was already present in the reduced set of practices identified from the first part of the survey. In spite of this, we wanted to make sure the subjects have had clear comprehension that it should be performed as part of the testing process.

The subjects were also asked whether a testing approach could be considered *fully defined* when the product risks were already analysed, and items and features to be tested were already defined. This question was motivated by the fact that ***Define the test approach*** (practice #5 in Fig. 7) was not present in the reduced set of practices derived from the initial questionnaire. For this question, we received 10 negative answers; that is, one cannot consider the testing approach fully defined only by analysing product risks and defining items and features to be tested. Therefore, we included the ***Define the test approach*** practice in the final set, thus resolving a dependency reported by Höhn (2011).

The third question of the second questionnaire addressed the ***Define exit criteria*** practice (#7 in Fig. 7), since it was not identified as mandatory after the first data analysis. Subjects were asked whether it is possible to run a test process without explicit exit criteria (*i.e.* information about when test should stop). Based on 9 negative answers (*i.e.* 65 %), this practice was also included in the reduced set.

This second analysis helped us to either clarify or resolve the aforementioned dependencies amongst TMMi practices. In the next sections we analyse and discuss the survey results. For this, we adapted Höhn's mind map (Höhn 2011) (Figs. 8, 9, 10, 11 and 12), according to each phase of a generic testing process. Practices highlighted in grey are identified as mandatory and should be implemented in any testing process.

Our analysis was also supported by the IEEE-829 Standard for Software and System Test Documentation (IEEE 2008). This standard presents a model for test plan and clearly indicates what this plan should contain. Maturity models present what should be done to complete a phase, but do not indicate what must be included in the documentation.

4.3.1 Planning

Planning the testing activity is definitely one of the most important process phases. It comprises the definition of how testing will be performed and what will be tested; it enables proper activity monitoring, control and measurement. The derived test plan includes details of the schedule, team, items to be tested, and the approach to be applied (IEEE 2008). In TMMi, planning-related practices also comprise non-functional testing, definition of the test environment and peer reviews. In total, 29 practices are related to planning (see Fig. 8), spread over the nine specific goals (labelled with *SG* in the figure).

To achieve these goals, the organisation must fulfil all the practices shown in Fig. 8. Despite this, our results show that only 8 out of these 29 practices are mandatory, according to the Complete Set subject group. According to Höhn's analysis, TMMi has internal dependencies amongst practices, some related to the Planning phase. Therefore, 2 other practices are necessary to resolve such dependencies (this is discussed in the sequence). Thus, the final set of 10 mandatory practices for the Planning phase is shown in grey background in Fig. 8.

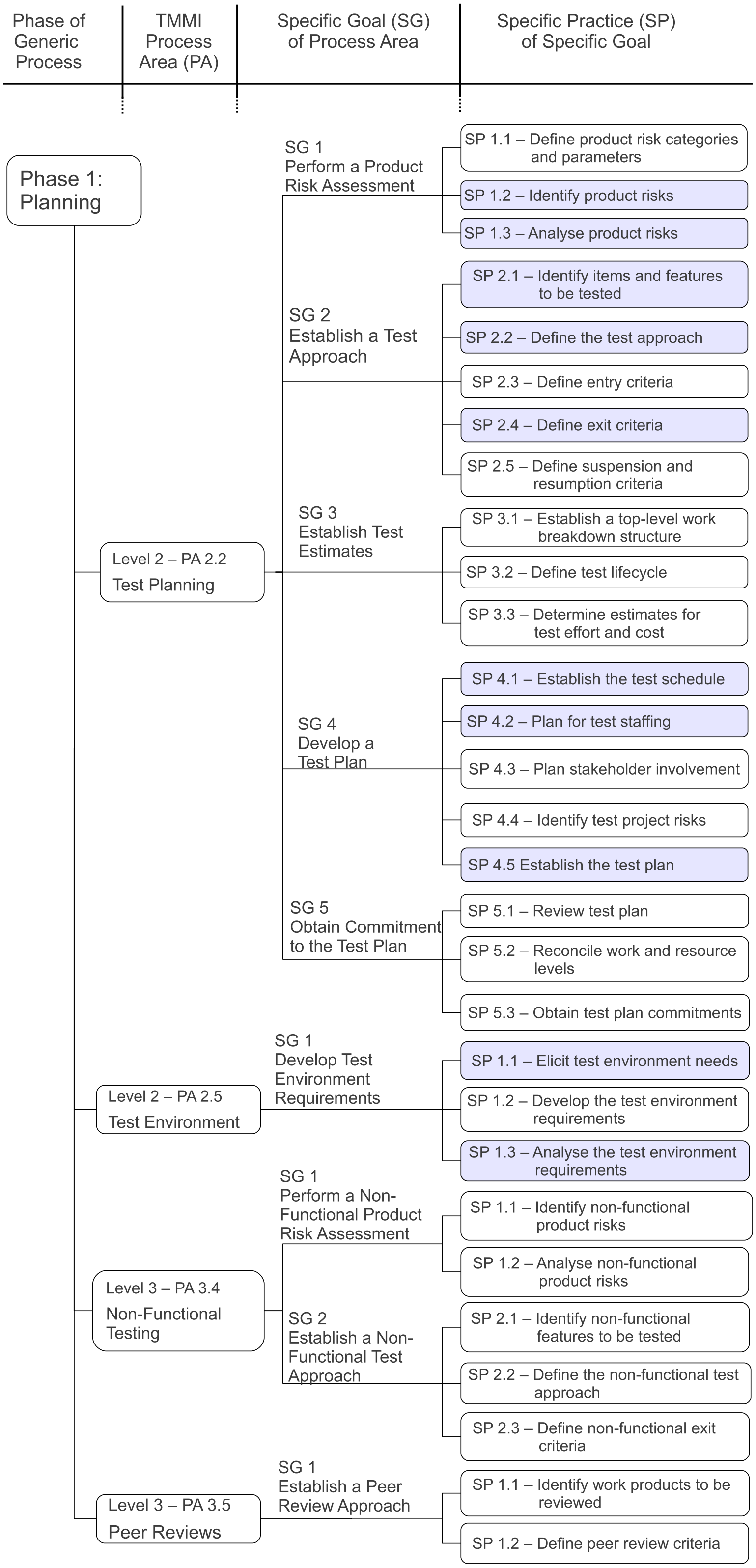

Phase of Generic Process
TMMI Process Area (PA)
Specific Goal (SG) of Process Area
Specific Practice (SP) of Specific Goal
Phase 1: Planning
Level 2 – PA 2.2
Test Planning
SG 1
Perform a Product Risk Assessment
SP 1.1 – Define product risk categories and parameters
SP 1.2 – Identify product risks
SP 1.3 – Analyse product risks
SG 2
Establish a Test Approach
SP 2.1 – Identify items and features to be tested
SP 2.2 – Define the test approach
SP 2.3 – Define entry criteria
SP 2.4 – Define exit criteria
SP 2.5 – Define suspension and resumption criteria
SG 3
Establish Test Estimates
SP 3.1 – Establish a top-level work breakdown structure
SP 3.2 – Define test lifecycle
SP 3.3 – Determine estimates for test effort and cost
SG 4
Develop a Test Plan
SP 4.1 – Establish the test schedule
SP 4.2 – Plan for test staffing
SP 4.3 – Plan stakeholder involvement
SP 4.4 – Identify test project risks
SP 4.5 Establish the test plan
SG 5
Obtain Commitment to the Test Plan
SP 5.1 – Review test plan
SP 5.2 – Reconcile work and resource levels
SP 5.3 – Obtain test plan commitments
Level 2 – PA 2.5
Test Environment
SG 1
Develop Test Environment Requirements
SP 1.1 – Elicit test environment needs
SP 1.2 – Develop the test environment requirements
SP 1.3 – Analyse the test environment requirements
Level 3 – PA 3.4
Non-Functional Testing
SG 1
Perform a Non-Functional Product Risk Assessment
SP 1.1 – Identify non-functional product risks
SP 1.2 – Analyse non-functional product risks
SG 2
Establish a Non-Functional Test Approach
SP 2.1 – Identify non-functional features to be tested
SP 2.2 – Define the non-functional test approach
SP 2.3 – Define non-functional exit criteria
Level 3 – PA 3.5
Peer Reviews
SG 1
Establish a Peer Review Approach
SP 1.1 – Identify work products to be reviewed
SP 1.2 – Define peer review criteria

Fig. 8 TMMi practices related to Planning

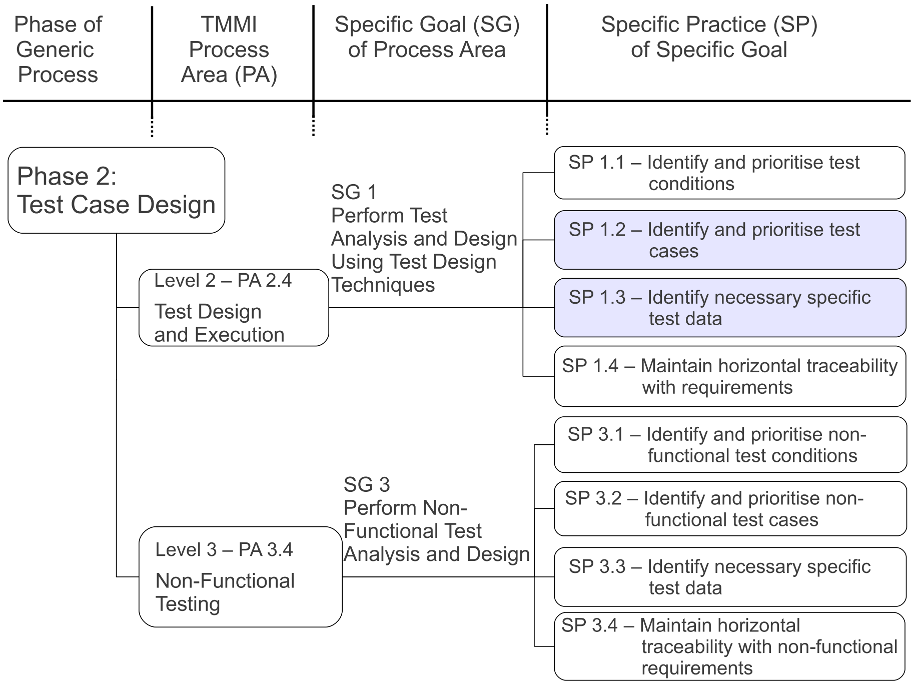

Phase of Generic Process
TMMI Process Area (PA)
Specific Goal (SG) of Process Area
Specific Practice (SP) of Specific Goal
Phase 2: Test Case Design
Level 2 – PA 2.4
Test Design and Execution
SG 1
Perform Test Analysis and Design Using Test Design Techniques
SP 1.1 – Identify and prioritise test conditions
SP 1.2 – Identify and prioritise test cases
SP 1.3 – Identify necessary specific test data
SP 1.4 – Maintain horizontal traceability with requirements
Level 3 – PA 3.4
Non-Functional Testing
SG 3
Perform Non-Functional Test Analysis and Design
SP 3.1 – Identify and prioritise non-functional test conditions
SP 3.2 – Identify and prioritise non-functional test cases
SP 3.3 – Identify necessary specific test data
SP 3.4 – Maintain horizontal traceability with non-functional requirements

Fig. 9 TMMi practices related to Test Case Design

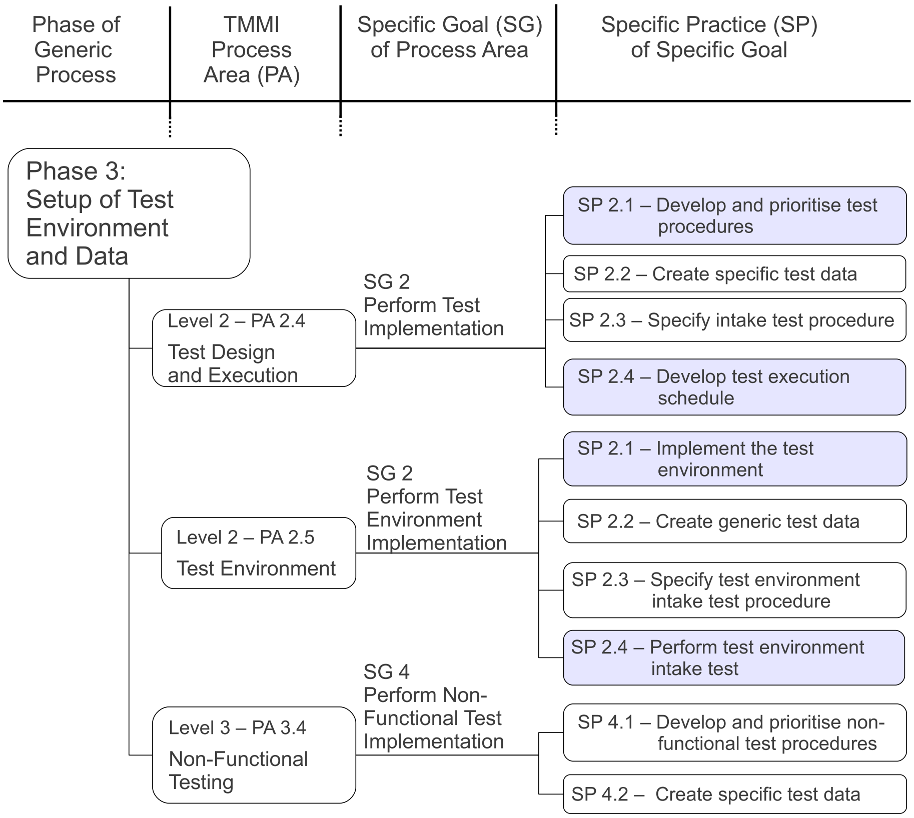

Phase of Generic Process
TMMI Process Area (PA)
Specific Goal (SG) of Process Area
Specific Practice (SP) of Specific Goal
Phase 3: Setup of Test Environment and Data
Level 2 – PA 2.4
Test Design and Execution
SG 2
Perform Test Implementation
SP 2.1 – Develop and prioritise test procedures
SP 2.2 – Create specific test data
SP 2.3 – Specify intake test procedure
SP 2.4 – Develop test execution schedule
Level 2 – PA 2.5
Test Environment
SG 2
Perform Test Environment Implementation
SP 2.1 – Implement the test environment
SP 2.2 – Create generic test data
SP 2.3 – Specify test environment intake test procedure
SP 2.4 – Perform test environment intake test
Level 3 – PA 3.4
Non-Functional Testing
SG 4
Perform Non-Functional Test Implementation
SP 4.1 – Develop and prioritise non-functional test procedures
SP 4.2 – Create specific test data

Fig. 10 TMMi practices related to Setup of Test Environment and Data

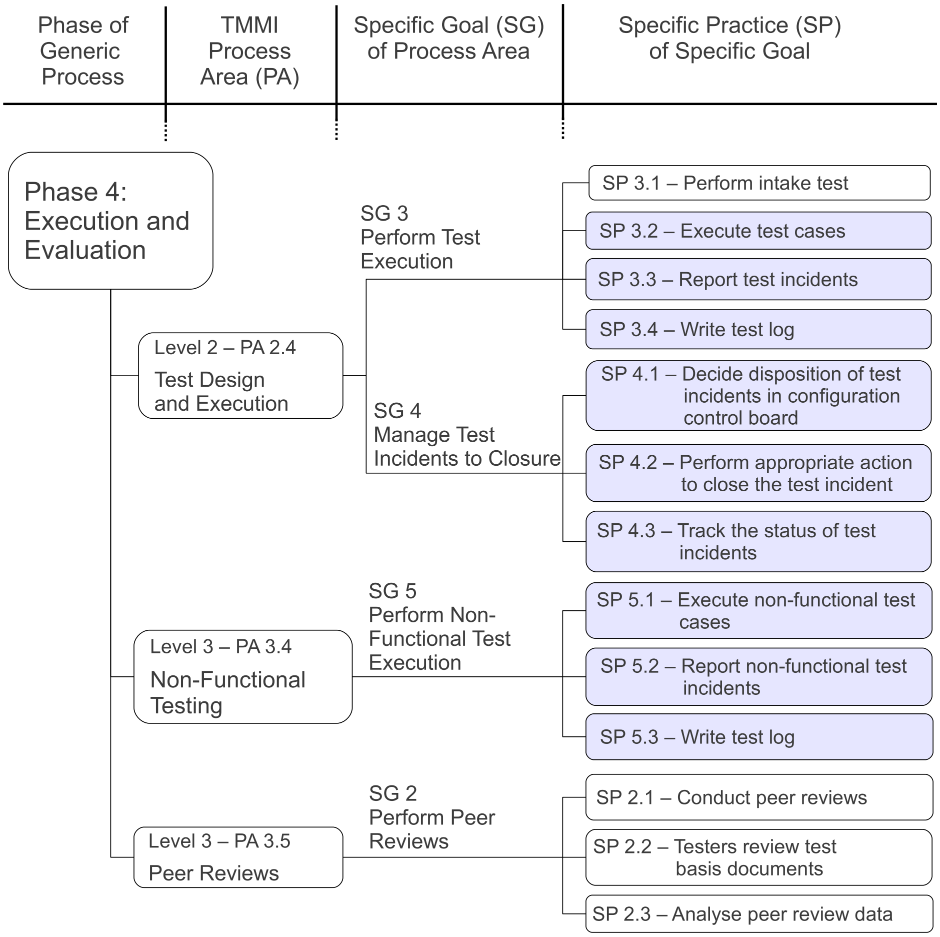

Phase of Generic Process
TMMI Process Area (PA)
Specific Goal (SG) of Process Area
Specific Practice (SP) of Specific Goal
Phase 4: Execution and Evaluation
Level 2 – PA 2.4
Test Design and Execution
SG 3
Perform Test Execution
SP 3.1 – Perform intake test
SP 3.2 – Execute test cases
SP 3.3 – Report test incidents
SP 3.4 – Write test log
SG 4
Manage Test Incidents to Closure
SP 4.1 – Decide disposition of test incidents in configuration control board
SP 4.2 – Perform appropriate action to close the test incident
SP 4.3 – Track the status of test incidents
Level 3 – PA 3.4
Non-Functional Testing
SG 5
Perform Non-Functional Test Execution
SP 5.1 – Execute non-functional test cases
SP 5.2 – Report non-functional test incidents
SP 5.3 – Write test log
Level 3 – PA 3.5
Peer Reviews
SG 2
Perform Peer Reviews
SP 2.1 – Conduct peer reviews
SP 2.2 – Testers review test basis documents
SP 2.3 – Analyse peer review data

Fig. 11 TMMi practices related to Execution and Evaluation

Amongst these practices, ***Identify product risks*** and ***Analyse product risks*** demonstrate the relevance of evaluating product risks. Their output plays key role in the testing approach definition and test case prioritisation. The product risks consist of a list of potential problems that should be considered while defining the test plan. Figure 7 shows that these two practices were mostly ranked as mandatory considering all profiles.

According to the IEEE-829 Standard for Software and System Test Documentation (IEEE 2008), a test plan should include: a list of what will be and will not be tested; the approach to be used; the schedule; the testing team; test classes and conditions; exit criteria etc. In our survey, ***Identify items and features to be tested***, ***Establish the test schedule*** and ***Plan for test staffing*** practices were mostly ranked as mandatory. They are directly related to ***Establish the test plan***, and address the definition of most of the items listed in the IEEE-829 Standard. This is complemented with the ***Define exit criteria***, selected after the dependency resolution. This evinces the coherence of the survey's subject choices for mandatory practices with respect to the Planning phase.

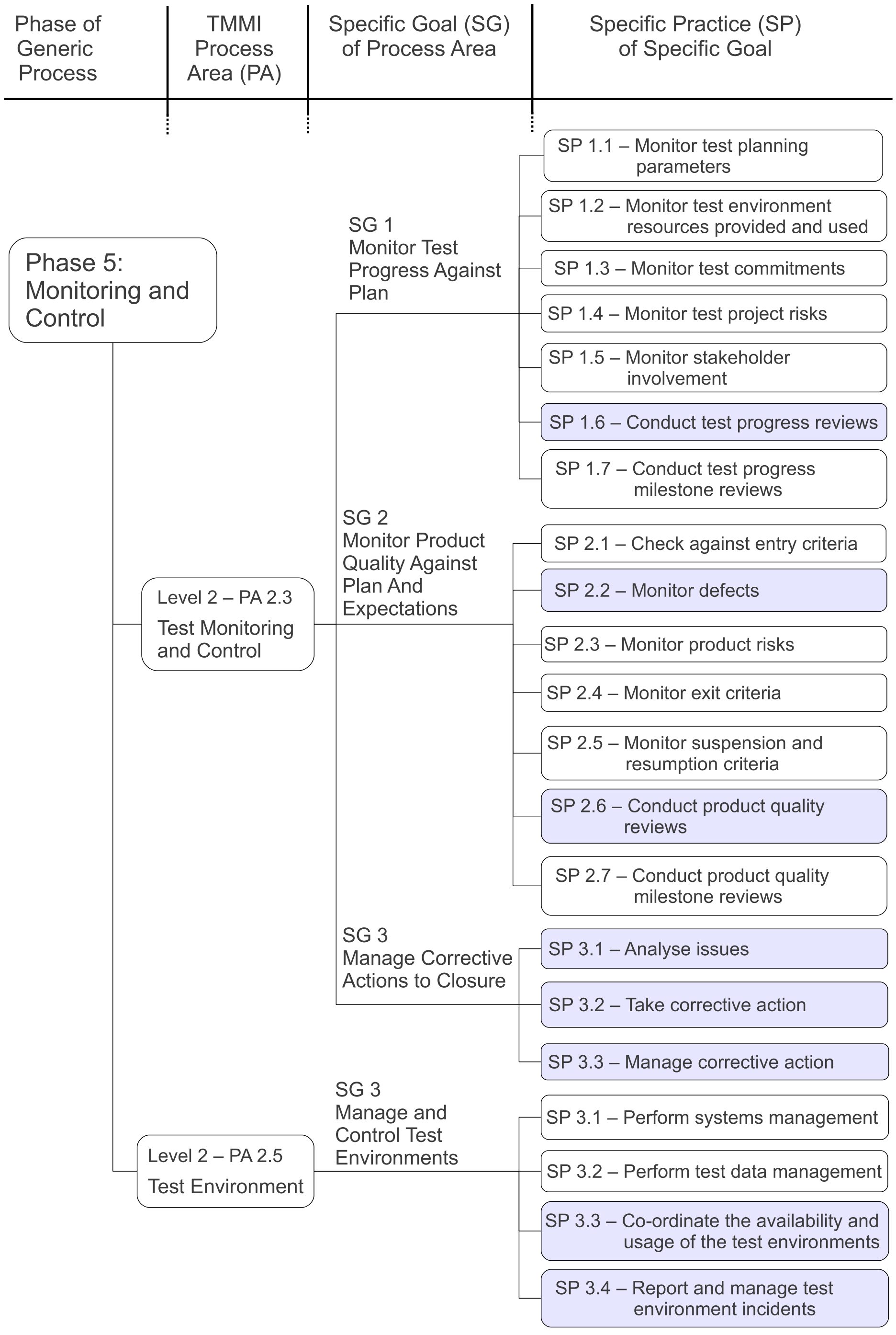

Phase of Generic Process
TMMI Process Area (PA)
Specific Goal (SG) of Process Area
Specific Practice (SP) of Specific Goal
Phase 5: Monitoring and Control
Level 2 – PA 2.3
Test Monitoring and Control
Level 2 – PA 2.5
Test Environment
SG 1 Monitor Test Progress Against Plan
SG 2 Monitor Product Quality Against Plan And Expectations
SG 3 Manage Corrective Actions to Closure
SG 3 Manage and Control Test Environments
SP 1.1 – Monitor test planning parameters
SP 1.2 – Monitor test environment resources provided and used
SP 1.3 – Monitor test commitments
SP 1.4 – Monitor test project risks
SP 1.5 – Monitor stakeholder involvement
SP 1.6 – Conduct test progress reviews
SP 1.7 – Conduct test progress milestone reviews
SP 2.1 – Check against entry criteria
SP 2.2 – Monitor defects
SP 2.3 – Monitor product risks
SP 2.4 – Monitor exit criteria
SP 2.5 – Monitor suspension and resumption criteria
SP 2.6 – Conduct product quality reviews
SP 2.7 – Conduct product quality milestone reviews
SP 3.1 – Analyse issues
SP 3.2 – Take corrective action
SP 3.3 – Manage corrective action
SP 3.1 – Perform systems management
SP 3.2 – Perform test data management
SP 3.3 – Co-ordinate the availability and usage of the test environments
SP 3.4 – Report and manage test environment incidents

Fig. 12 TMMi practices related to Monitoring and Control

The Planing phase also includes practices that address the definition of the test environment. In regard to this, ***Elicit test environment needs*** and ***Analyse the test environment requirements*** are ranked as mandatory and as clearly inter-related.

To conclude this analysis regarding the Planning phase, note that not all TMMi specific goals are achieved only with the execution of this selection of mandatory practices.

Despite this, the selected practices are able to yield a feasible test plan and make the process clear, managed and measurable.

After Planning, the next phase is related to Test Case Design. The input to this phase if the test plan, which includes some essential definitions such as risk analysis, the items which will be tested and the adopted approach.

4.3.2 Test case design

Figure 9 summarises the results of our survey for this phase, based on the set of TMMi practices identified by Höhn (2011). As the reader can notice, only two practices were mostly ranked as mandatory by the Complete Set group of subjects: ***Identify and prioritise test cases*** and ***Identify necessary specific test data*** (both shown in grey background in Fig. 9).

According to the IEEE-829 Standard, the test plan encompasses some items related to test case design, such as the definition of test classes and conditions (IEEE 2008). Due to this, it is likely that part of the subjects considers that the test plan itself already fulfils the needs regarding test case designing, thus most of the practices are not really necessary. For instance, if we considered solely the Profile-MR-MPS, none of the practices within this phase would appear in the results (see Fig. 7 to double-check this finding). On the other hand, subjects of the other profiles consider some other practices of this phase should be explicitly performed in a testing process. For instance, subjects of the Profile-Specialist profile ranked ***Identify and prioritise test conditions***, ***Identify necessary specific test data*** and ***Maintain horizontal traceability with requirements*** as mandatory. For the Profile-TMMi subjects, ***Identify and prioritise test cases*** and ***Maintain horizontal traceability with requirements*** should be mandatory.

From these results, we can conclude that there is uncertainty about what should indeed be done during the test case design phase. Moreover, this uncertainty may also indicate that not always test cases are documented separately from the test plan. From our observations in the industry context, a common practice is not to have a clear phase for test case designing, in general due to time constraints. The planning phase usually includes the designing of test. So, it is reasonable that the test plan itself includes the test cases, testing approach (and its underlying conditions) and the exit criteria. Thus, the two selected practices for this phase complement the needs to compose a feasible, streamlined testing process.

4.3.3 Setup of test environment and data

As discussed in Section 4.3.1, in the Planning phase test environment requirements are identified and described. The Setup of Test Environment and Data phase addresses the prioritisation and implementation of such requirements. Figure 10 shows the TMMi specific goals and practices for this phase.

According to TMMi, ***Develop and prioritise test procedures*** consists in determining the order test cases will be executed. Such order is defined in accordance with the product risks. The classification of this practice as mandatory is aligned with the practices selected for the Planning phase, some of which related to risk analysis. Another practice ranked as mandatory is ***Develop test execution schedule***, which is directly related to the prioritisation of test case execution. The other two practices (*i.e.* ***Implement the test environment*** and ***Perform test environment intake test***) address the environment implementation and ensuring it is operational, respectively. The conclusion regarding this phase is that the four practices are sufficient to create an adequate environment to run the tests.

4.3.4 Execution and evaluation

The next phase of a generic testing process consists of test case execution and evaluation. At this point, the team runs the tests and, eventually, creates the defect reports. The evaluation aims to assure the test goals were achieved and to inform the results to stakeholders (Hass 2008. For this phase, Höhn (2011) identified 13 TMMi practices, which are related to test execution goals, management of incidents, non-functional test execution and peer reviews. This can be seen in Fig. 11. As the reader can notice, only four practices *were not* ranked as mandatory. This makes evident the relevance of this phase, since it encompasses activities which are related to test execution and management of incidents.

The results summarised in Fig. 11 include practices that regard the execution of non-functional tests. However, in the Planning an Test Case Design phases, the selected practices do not address the definition of such type of tests. Although this sounds incoherent, this may indicate that, from the planning and design viewpoints, there is not a clear separation between functional and non-functional testing. The separation is a characteristic of the TMMi structure, but for the testing community these two types of testing are performed in conjunction, since the associated practices as described in TMMi are very similar in both cases.

4.3.5 Monitoring and control

The execution of the four phases of a generic testing process yields a substantial amount of information. Such information needs to be organised and consolidated to enable rapid status checking and, if necessary, corrective actions. This is addressed during the Monitoring and Control phase (Crespo et al. 2010).

Figure 12 depicts the TMMi practices with respect to this phase. Again, the practices ranked as mandatory by most of the subjects are highlighted in grey. Note that there is consensus amongst all profile groups (*i.e.* Profile-Specialist, Profile-MR-MPS, Profile-TMMi and the Complete Set) about what is mandatory regarding Monitoring and Control. This can be cross-checked in Fig. 7.

Performing the ***Conduct test progress reviews*** and ***Conduct product quality reviews*** practices means keeping track of both the testing process status and the product quality, respectively. ***Monitor defects*** addresses gathering metrics that concern incidents (also referred to as *issues*), while ***Analyse issues***, ***Take corrective action*** and ***Manage corrective action*** are clearly inter-related practices. The two other practices considered mandatory within this phase are ***Co-ordinate the availability and usage of the test environments*** and ***Report and manage test environment incidents***. Both are important since either unavailability or incidents in the test environment may compromise the activity as a whole.

As a final note with respect to the survey results, we emphasise that the subjects were not provided with any information about dependencies amongst TMMi practices. Besides this, we were aware that the inclusion of practices not mostly ranked as mandatory might have been created new broken dependencies. Despite this, the analysis of the final set of mandatory practices shows that all dependencies are resolved.

5 Discussion

In this section we discuss the results of our survey – i.e. the streamlined testing process – in regard to the TMMi level which can be achieved with the process. The goal of this discussion is verifying which TMMi level a software company that fully runs the reduced process may achieve in case such company is pursuing a TMMi certification. As described in Section 3, we used the KITTool tool (Höhn 2011) to support the data collection.

The spider chart depicted in Fig. 13 provides an overall view of the diagnosis obtained with KITTool. It is evident that the focus is on ***PA2.4 – Test Design and Execution***. This may reflect the state of the practice of a sample of the Brazilian software industry. Companies concentrate their efforts on the design and execution of tests, whilst less attention is paid on test planning, monitoring and control.

Figure 14 depicts a more fine-grained view of the diagnosis. As the reader can notice, most of fulfilled practices regard the *Managed* level of TMMi, *i.e.* level 2. In total, 26 % (19 out of 71) practices are defined, documented and implemented if we consider the streamlined testing process. The Process Area (PA) which is mostly addressed is ***PA2.4 – Test Design and Execution***; such PA is 48 % fulfilled with the obtained process. On the other hand, ***PA2.2 – Test Planning*** is the least addressed PA; only 19 % of its practices are implemented.

The results depicted in Fig. 14 reveal some interesting insights. Firstly, the "ideal" process, as seen by experienced Brazilian testing professionals (*i.e.* the survey subjects), focuses on practical matters like test design and execution, including practices related to test prioritisation, test data definition and incident management. Other not less important tasks, on the other hand, are left in a second plan; examples are test environment preparation (28 % fulfilled) and monitoring and control (37 % fulfilled).

Once more we highlight the streamlined process characterised in this paper represents a consistent process. This was demonstrated with the dependency analysis presented at the beginning of Section 4.3. Nonetheless, if a software company aims to ascend TMMi levels, the diagnosis presented in this section may be used as a reference for the introduction of new practices towards reaching such an objective.

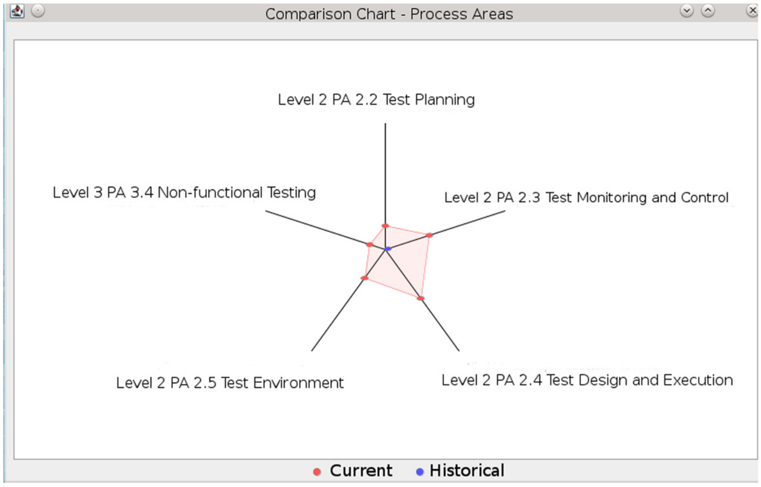
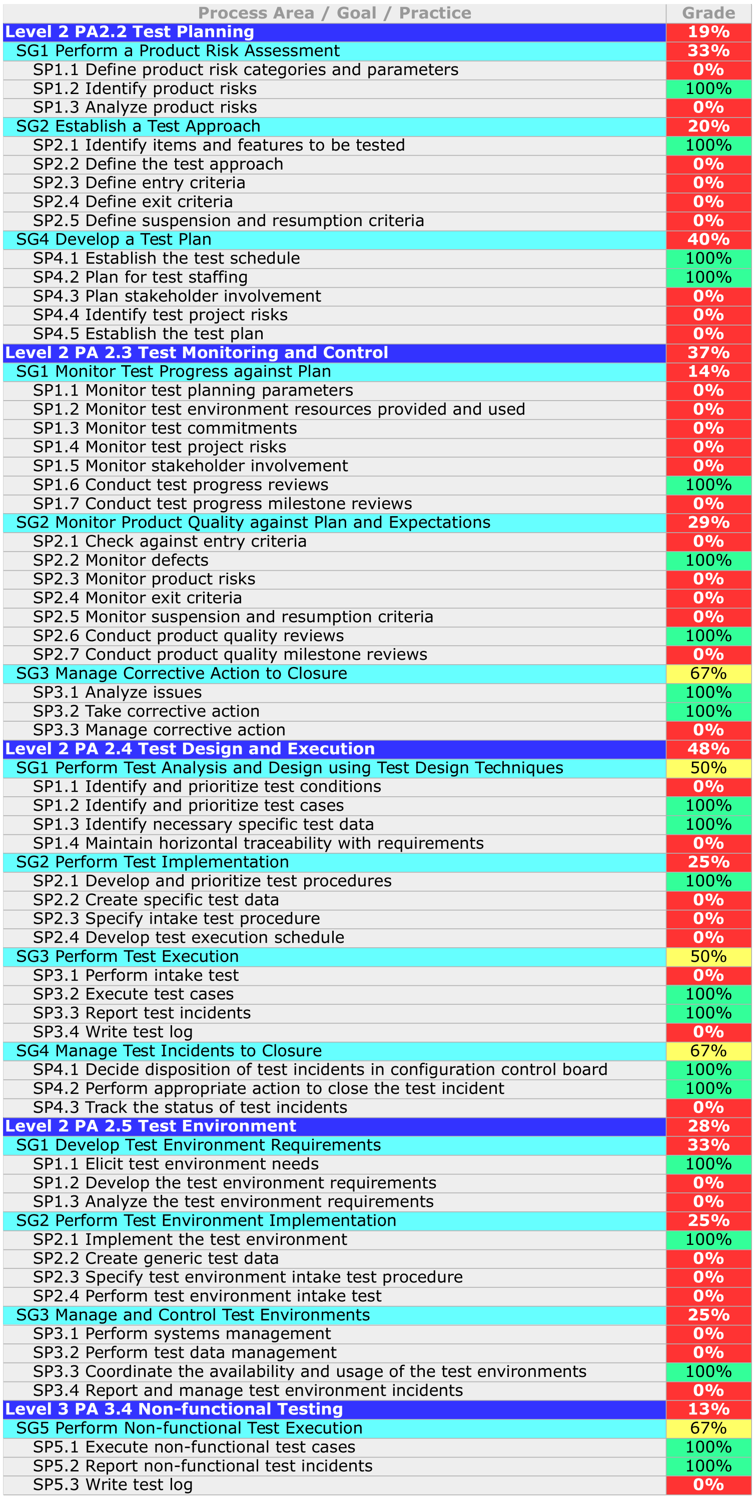


Fig. 13 Fulfilment of TMMi Process Areas with the streamlined process adoption



Process Area / Goal / Practice	Grade
Level 2 PA2.2 Test Planning	**19%**
SG1 Perform a Product Risk Assessment	**33%**
SP1.1 Define product risk categories and parameters	**0%**
SP1.2 Identify product risks	100%
SP1.3 Analyze product risks	**0%**
SG2 Establish a Test Approach	**20%**
SP2.1 Identify items and features to be tested	100%
SP2.2 Define the test approach	**0%**
SP2.3 Define entry criteria	**0%**
SP2.4 Define exit criteria	**0%**
SP2.5 Define suspension and resumption criteria	**0%**
SG4 Develop a Test Plan	**40%**
SP4.1 Establish the test schedule	100%
SP4.2 Plan for test staffing	100%
SP4.3 Plan stakeholder involvement	**0%**
SP4.4 Identify test project risks	**0%**
SP4.5 Establish the test plan	**0%**
Level 2 PA 2.3 Test Monitoring and Control	**37%**
SG1 Monitor Test Progress against Plan	**14%**
SP1.1 Monitor test planning parameters	**0%**
SP1.2 Monitor test environment resources provided and used	**0%**
SP1.3 Monitor test commitments	**0%**
SP1.4 Monitor test project risks	**0%**
SP1.5 Monitor stakeholder involvement	**0%**
SP1.6 Conduct test progress reviews	100%
SP1.7 Conduct test progress milestone reviews	**0%**
SG2 Monitor Product Quality against Plan and Expectations	**29%**
SP2.1 Check against entry criteria	**0%**
SP2.2 Monitor defects	100%
SP2.3 Monitor product risks	**0%**
SP2.4 Monitor exit criteria	**0%**
SP2.5 Monitor suspension and resumption criteria	**0%**
SP2.6 Conduct product quality reviews	100%
SP2.7 Conduct product quality milestone reviews	**0%**
SG3 Manage Corrective Action to Closure	67%
SP3.1 Analyze issues	100%
SP3.2 Take corrective action	100%
SP3.3 Manage corrective action	**0%**
Level 2 PA 2.4 Test Design and Execution	**48%**
SG1 Perform Test Analysis and Design using Test Design Techniques	50%
SP1.1 Identify and prioritize test conditions	**0%**
SP1.2 Identify and prioritize test cases	100%
SP1.3 Identify necessary specific test data	100%
SP1.4 Maintain horizontal traceability with requirements	**0%**
SG2 Perform Test Implementation	**25%**
SP2.1 Develop and prioritize test procedures	100%
SP2.2 Create specific test data	**0%**
SP2.3 Specify intake test procedure	**0%**
SP2.4 Develop test execution schedule	**0%**
SG3 Perform Test Execution	50%
SP3.1 Perform intake test	**0%**
SP3.2 Execute test cases	100%
SP3.3 Report test incidents	100%
SP3.4 Write test log	**0%**
SG4 Manage Test Incidents to Closure	67%
SP4.1 Decide disposition of test incidents in configuration control board	100%
SP4.2 Perform appropriate action to close the test incident	100%
SP4.3 Track the status of test incidents	**0%**
Level 2 PA 2.5 Test Environment	**28%**
SG1 Develop Test Environment Requirements	**33%**
SP1.1 Elicit test environment needs	100%
SP1.2 Develop the test environment requirements	**0%**
SP1.3 Analyze the test environment requirements	**0%**
SG2 Perform Test Environment Implementation	**25%**
SP2.1 Implement the test environment	100%
SP2.2 Create generic test data	**0%**
SP2.3 Specify test environment intake test procedure	**0%**
SP2.4 Perform test environment intake test	**0%**
SG3 Manage and Control Test Environments	**25%**
SP3.1 Perform systems management	**0%**
SP3.2 Perform test data management	**0%**
SP3.3 Coordinate the availability and usage of the test environments	100%
SP3.4 Report and manage test environment incidents	**0%**
Level 3 PA 3.4 Non-functional Testing	**13%**
SG5 Perform Non-functional Test Execution	67%
SP5.1 Execute non-functional test cases	100%
SP5.2 Report non-functional test incidents	100%
SP5.3 Write test log	**0%**

Fig. 14 Testing process diagnosis supported by KITTool

6 Validity threats

This section describes some issues that may threaten the validity of our results. Despite this, the study limitations did not prevent the achievement of significant results with respect to software testing process definition, based on the opinion of software testing professionals.

A first limitation concerns the questionnaire design. The questions were based on the TMMi structure, so were the help notes provided together with the questions. Even though the intent of the help notes was facilitating the subjects' understanding regarding the questions, they might not have been enough to allow for correct comprehension. For instance, in this survey it was clear that the practices related to functional and non-functional testing were not understood as distinct activities, since they were ranked as mandatory only in the Execution and Evaluation phase.

Still regarding the survey design, previous knowledge on the TMMi structure (or even on other maturity models) also represents a validity threat with respect to construction (Wohlin et al. 2000). During the survey, this threat could not be avoided, since we intended to gather to opinion of experienced testing professionals.

Another threat regards the scale of values used in the first questionnaire. The answer scale was composed of four values. This represented a limitation for the statistical analysis, since the responses were mostly concentrated in values 3 and 4. If a wider scale were used, *e.g.* from 1 to 10, this could have yielded a better distribution of answers, thus enabling us to apply a more adequate interpretation model.

The sample size was also a limitation of the study. In practice, although the sample includes only software testing professionals, its size is reduced in the face of the real population. Perhaps the way the participation call was announced and the time it was available have limited the sample.

Last but not least, there are some other standardised testing processes and maturity models (*e.g.* ISO/IEC/IEEE 29119 (SoftwareTestingStandard.org 2014) and MPT.Br (Softex Recife 2011)) that might also be used as a baseline for the definition of a streamlined process. For instance, the ISO/IEC/IEEE 29119 standard splits the process into a more fine-grained set of phases. Despite this, there is clear overlapping between the sets of TMMi practices and activities from these standards and models, which we believe contributes to the fairness of the industry observation. We highlight that a similar approach as the one described in our paper could be followed to reduce the number of activities of other testing process standards and models.

7 Conclusions

This paper described a survey which was conducted in two stages and investigated whether there is a subset of TMMi practices that can be considered essential for a generic testing process. The survey was applied amongst professionals who work with software testing. The results were reported in our previous work (Camargo et al. 2013). To extend our research, this paper also described the results of an additional analysis regarding the TMMi level that might be achieved with the characterised process.

The analysis of the survey results led us to conclude that, from the set of 81 TMMi practices distributed by Höhn (2011) across the phases of a generic testing process, 33 are considered essential for maintaining consistency when such a process is defined. This represents a reduction of around 60 % in the number of TMMi practices. Note that the other TMMi practices are not disposable; however, when the goal is to implement a streamlined process, or even when the company does not have the necessary know-how to implement its own testing process, it can use this reduced set of practices to do so. Thus, the results reported in this paper represent a simplified way to create or improve testing processes, which is based on a recognised reference model.

The practices highlighted in Figs. 8, 9, 10, 11 and 12 can also indicate the priority of implementation for a company that is using TMMi as a reference for its testing process. This model does not indicate what can be implemented first, or the possible dependencies amongst the process areas. Nonetheless, the results of this study point out a set of activities that can be implemented as a priority. At a later stage, a company may decide to continue to deploy the remaining practices required by the model in order to obtain the TMMi certification.

TMMi is fine-grained in terms of practices and their distribution across the specific goals and process areas. Even though this may ease the implementation of practices, this makes the model complex and difficult to understand. Once a company is willing to build a testing process based on a reference model, this process must be in accordance with its reality. Not all TMMi practices are feasible for all sizes of companies and teams. The results, analyses and discussions presented in this paper makes this conclusion more evident: the TMMi practices that are considered more relevant for testing experts mostly address the level 2 of this maturity model. The selected set of practices is far from fulfilling 100 % of any Process Area.

Thus, it is important to be aware of a basic set of practices that, if not performed, may compromise the quality of the process defined for a given context (*e.g.* for a particular company or project), and hence the quality of the product under test. In this sense, we hope the results of this work can support small and medium companies that wish to implement a new testing process, or even improve their current processes.

Endnote

[1] Note that we had a total of 37 set of answers for 81 questions; thus, the four groups shown in Fig. 5 sum up 2997 individual answers.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

KC was responsible for the study planning and setup, data collection and analysis and the manuscript preparation. FF was responsible for study planning and setup, data analysis and the manuscript preparation. SF was responsible for study planning, data analysis and the manuscript preparation. All authors read and approved the final manuscript.

Acknowledgements

The authors are grateful for the financial support provided by CAPES and CNPq, Brazil.

Received: 3 June 2014 Accepted: 8 May 2015
Published online: 02 June 2015

References

Andersin J (2004) TPI- a model for test process improvement. Seminar. University of Helsinki, Helsinki - Finland

Basili VR, Reiter RW (1981) A controlled experiment quantitatively comparing software development approaches. IEEE Trans Softw Eng SE-7(3):299–320

Cao P, Dong Z, Liu K (2010) An optimal release policy for software testing process. In: Proceedings of the 29th Chinese Control Conference (CCC). IEEE, Beijing - China. pp 6037–6042

Camargo KG (2012) Test Process Survey. http://amon.dc.ufscar.br/limesurvey/index.php?sid=47762&lang=pt-BR. Accessed on 18/05/2015

Camargo KG, Ferrari FC, Fabbri SCPF (2013) Identifying a subset of TMMi practices to establish a streamlined software testing process. In: Proceedings of the 27^{th} Brazilian Symposium on Software Engineering (SBES). IEEE, Brasília/DF - Brazil. pp 137–146

Crespo AN, Jino M, Argollo M, Bueno PMS, Barros CP (2010) Generic Process Model for Software Testing. Online. http://www.softwarepublico.gov.br/5cqualibr/xowiki/Teste-item13 - accessed on 19/12/2014 (in Portuguese)

Experimentus Ltd. (2012) Test Maturity Model integrated (TMMi) Survey results. Online. http://www.tmmi.org/pdf/tmmisurvey2012.pdf - Accessed on 19/12/2014

Hass AMJ (2008) Testing processes. In: IEEE International Conference on Software Testing Verification and Validation Workshop (ICSTW). IEEE, Lillehammer, Norway. pp 321–327

Höhn EN (2011) KITest: A framework of knowledge and improvement of testing process. PhD thesis. University of São Paulo, São Carlos, SP - Brazil (in Portuguese)

IEEE (2008) IEEE Standard for Software and System Test Documentation, Institute of Electric and Electronic Engineers, Std 829-2008. IEEE Computer Society, New York, NY, USA

LimeSurvey (2014). online. http://www.limesurvey.org/ - Accessed on 19/12/2014

Melo C (2004) DFTestes Mailing List. http://br.dir.groups.yahoo.com/group/DFTestes. Accessed on 18/05/2015

Miller J (2004) Statistical significance testing: a panacea for software technology experiments? J Syst Softw 73(2):183–192

PISO (2015) Polo Industrial de Software. http://www.piso.org.br/. Accessed on 18/05/2015

Purper CB (2000) Transcribing process model standards into meta-processes. In: Proceedings of the 7th European Workshop on Software Process Technology (EWSPT) - Lecture Notes in Computer Science V.1780. Springer, Kaprun - Austria. pp 55–68

Rodrigues A, Pinheiro PR, Albuquerque A (2010) The definiton of a testing process to small-sized companies: The Brazilian scenario. In: Proceedings of the 7th International Conference on the Quality of Information and Communications Technology (QUATIC). IEEE, Porto - Portugal. pp 298–303

SEI (2006) Capability Maturity Model Integration Version 1.2. (CMMI-SE/SW, V1.2 – Continuous Representation). Tech. Report CMU/SEI-2006-TR-001, Carnegie Mellon University

Softex Recife (2011) Model Reference Guide - MPT.Br (in Portuguese). Online. http://mpt.org.br/mpt/wpcontent/uploads/2013/05/MPT_Guia_de_referencia.pdf - Accessed on 19/12/2014

Softex (2012) Improvement of Brazilian Software Process - General Guide (in Portuguese). Softex - Association for Promoting Excellency in Brazilian Software, MR-MPS-SW 2012/. Softex, Recife - Brasil

SoftwareTestingStandard.org (2014) ISO/IEC/IEEE 29119 Software Testing - The International Standard for Software Testing. Online. http://softwaretestingstandard.org/ - Accessed on 19/12/2014

TMMI Foudation (2012) Test Maturity Model integration (TMMi) (Release 1). Online. http://www.tmmi.org/pdf/TMMi.Framework.pdf - Accessed on 19/12/2014

Whitley E, Ball J (2002) Statistics review 6: Nonparametric methods. Crit Care 6(6):509

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2000) Experimentation in Software Engineering: an Introduction. Kluwer Academic Publishers, Norwell, MA - USA