An Empirical Study on the Knowledge Management Practice in Software Testing

Claudia P. C. Maciel[1], Érica Ferreira de Souza[1], Nandamudi L. Vijaykumar[2], Ricardo de Almeida Falbo[3], Giovani Volnei Meinerz[1], and Katia Romero Felizardo[1]

Department of Computer, Federal Technological University of Paraná – UTFPR[1]
Cornélio Procópio/PR - Brazil
claudiamaciel@alunos.utfpr.edu.br,{ericasouza,giovanimeinerz, katiascannavino}@utfpr.edu.br
Lab. of Comp. and Applied Math., National Institute for Space Research – INPE[2]
São José dos Campos/SP - Brazil
vijay.nl@inpe.br
Depart. of Computer Science, Federal University of Espírito Santo – UFES[3]
Vitória/ES, Brazil
falbo@inf.ufes.br

Abstract. Context: Software testing is a knowledge intensive process, and therefore can benefit from the use of experience gained from past projects. In this context, principles of Knowledge Management (KM) can be applied to promote knowledge capture and sharing as well as the emergence of new knowledge. In spite of this, there are only few studies that present practitioners' perspective about KM initiatives in software testing. **Objective:** The main goal of this study is to identify the perception of professionals in software testing on the use of KM initiatives in software engineering companies, such as KM adoption, potential benefits or hindrances. **Method:** A survey was conducted in software development companies. In addition, the survey results were compared with another survey conducted previously in the same context, however, with different objectives and target audience. 39 software companies participated in this research. **Results:** Testing planning activity and the test case reuse are the aspects that have received more attention by companies. A strategic planning and reuse of existing test cases have a potential to significantly reduce software development costs and time. **Conclusion:** Applying KM in software companies can bring several benefits in terms of quality of results, reduction costs, time and effort. However, it is still a challenge to be faced and tackled.

1 Introduction

A major Software Engineering challenge is the creation of systematic strategies to integrate the knowledge of the actors involved in a project. Software organizations seek solutions that give an emphasis on different knowledge types when planning initiatives to manage them [1]. According to Levy and Hazzan

[2], a systematic routine to capture knowledge is important in an organization, making the knowledge transparent for all those involved, since without it an organization cannot benefit from its accumulated knowledge. However, although the importance of working with generated knowledge has been widely recognized in several areas, few organizations are able to manage such a generated knowledge. Knowledge Management (KM) principles are mechanisms that can solve, at least partially, such problems [3].

KM is concerned with how to capture, store and reuse dispersed knowledge in an environment [4]. Due to the importance of the developing products and services in companies, KM has become a promising area for conducting research and analysis to show evidence on its benefits [5]. Software Engineering, for instance, has concentrated efforts on how to efficiently conduct KM, since it facilitates the knowledge flow and use in all phases of the entire software development process. Software processes generate a significant amount of information, and KM entails formally managing this information to facilitate access and reuse, as well as transfer best practices in the organization to build competitive advantage and cost reduction through the appropriation from knowledge [3].

One important subarea of Software Engineering is software testing. Software testing activities are strategic elements for conducting development of projects and ensuring product quality. However, with technological advancement and the emergence of increasingly critical applications, testing activity has become a complex task that can generate a lot of important information to be manipulated [3]. Software testing is a knowledge intensive process. Past experiences and knowledge gained can be reused, for example, in test case design, or in the recognition of faults [6].

Software testing can produce an important intellectual capital and thus new projects can benefit from the experience acquired with past projects. Given the great importance of knowledge for software testing and the potential benefits in managing this knowledge, several studies have recommended the application of KM in software testing [3]. However, there are only few studies that present practitioners' perspective from the area.

This paper aims to provide the perception of software testing professionals about the practical application of KM in software testing, such as KM adoption, potential benefits or hindrances in some testing process activities. We intend to provide an insight on the actual benefit of applying KM in software testing activities, in the professionals' vision, that may serve as a practical help for companies that want to conduct initiatives of such type, specially, novice companies.

In order to achieve the objective of this study, we adapted and reapplied the survey conducted by Souza et al. [7]. In [7], the goal was to identify the best scenario to apply KM in software testing. In that survey, the authors made available an online questionnaire within research groups and social networks, which was answered by participants with different profiles, among them, professionals, graduate students, professors and researchers with some experience in software testing. 86 participants answered the survey. The main difference in relation to Souza et al. [7] and our survey is the objectives and target audience.

In the survey presented in this paper, we analyze only answers from software testing professionals working in Brazilian software development companies. We conducted a survey with 39 Brazilian software development companies that systematically conduct software testing activities in their projects. From the 39 companies, 75 employees answered the survey. In addition, the results of this study were compared with the results from Souza et al. [7].

The remainder of this paper is organized as follows. Section 2 presents the main concepts related to the background of this paper. Section 3 presents the research planning, the main results and discussions. Section 4 presents related works. Finally, Section 5 presents the final considerations, as well as the potential threats to the study validity.

2 Background

Software testing consists of dynamic verification activities that analyze the behavior of a program in relation to its expected behavior. Testing activities can be supported by a well-defined process [8]. In general, a testing process consists of following activities: Test Planning, Test Case Design, Test Execution, and Test Results Analysis [8]. In the planning, the test environment for the project, planning and scheduling testing activities are defined. All test planning should be documented in a Test Plan. The test case design activity is intended to design, document, and implement test cases. During the execution activity, the test cases are run and the results are analyzed to determine whether the test cases can provide a verdict of passed or failed. In addition to the process, software testing involves, testing levels, artifacts, techniques, procedures, resources and tools that seek to control and organize tests to achieve high quality software.

Since testing activities have attracted significant attention from both industry and academia, different improvement models have been used to guide testing. However, during software testing, a large amount of information is processed and generated, and such models alone are not sufficient to improve the organizational testing process. It is still necessary to provide automated support for tasks of acquisition, processing, analysis and dissemination of test knowledge for reuse [3]. In this context, Knowledge Management (KM) can be used to manage knowledge of software testing [3, 7].

KM became more present within organizations in the 1980s, coming from the need to gain knowledge from the vast amount of information to develop new products, processes, and more flexible organizational arrangements, providing a sustainable competitive advantage [9]. Having control and ease of access to knowledge has now become a competitive and economic differential for organizations. According to Markkula [10], knowledge is information combined with experience, interpretation and reflection. It is a form of information ready to be applied in decision making.

Knowledge can be classified into two types: tacit and explicit [11]. Tacit knowledge is the subjective and experience-based knowledge, that cannot be documented, and typically remains only in people's mind. This type of knowledge

depends on personal experience and involves intangible factors such as beliefs, perspectives, values and intuition. Explicit knowledge, in turn, represents the objective and rational knowledge that can be documented, and thus it can be accessed by multiple people. The conversion of tacit to explicit knowledge offers greater value to the organization as it leads to the creation of new knowledge [4].

The major challenge to manage knowledge is that most of the knowledge in Software Engineering is tacit and hard to dedicate efforts to make it explicit. When tacit knowledge is transformed into explicit, the organization begins to enable the dissemination of strategic knowledge [4]. In the context of software testing, KM can be used to capture knowledge and experience generated during the testing process. However, generally, this knowledge has been stored in documents or in people's minds. Unfortunately, paper has limited accessibility and it is difficult to be updated. On the other hand, tacit knowledge can easily be lost when experienced individuals leave the organization. This is the reason that knowledge in tests should be collected systematically through systems or applications, stored in an organizational and shared repository throughout the organization. This fact encouraged the development of this research in trying to understand and to make evident the perception of experts in the software testing area on the application of KM initiatives in Brazilian software development companies.

3 Survey on Knowledge Management Practice in Software Testing

In order to analyze the professionals' perspective with respect to the application of KM initiatives in software testing, we conducted a survey with 39 Brazilian software development companies. A survey is an empirical research strategy for the collection of information from heterogeneous sources. It has been used to provide a general picture of technology usage in the software industry [12]. According to Kitchenham and Pfleeger [13], a survey is conducted to identify the opinion of a specific population, for example, about a new method, process, tool or technique that has been used for a while. A survey can be conducted following a process with the following phases [13]: (i) definition of the research objective; (ii) identification of the target audience that will respond to the survey; (iii) survey design; (iv) survey evaluation; (v) survey application; and (vi) results analysis and presentation.

Research objective: the main objective of this survey is to provide insight from software testing professionals on the practical application of KM in software testing.

Target audience: 39 software development companies were invited to participate in this survey. The authors contacted the companies directly inviting them to participate in the survey. Only companies that develop software and conduct some software testing process were invited. Thus, the sampling method was non-probabilistic. For privacy reasons, names of the companies are not mentioned

here. However, some of their characteristics are: the companies are located in Brazil; there are micro, small, medium and large software companies; the main products developed include critical systems, such as electronic scales receipt and metrology, customized systems to meet the needs of customers in several segments (aviation, pharmaceuticals, automotive parts), hospital systems and Enterprise Resource Planning (ERP). Companies specialized in software testing that has as main service the outsourcing of software testing also participated.

Survey project: We used a questionnaire with sixteen questions divided into two parts. The first part of the questionnaire includes questions to identify the participant profile, such as experience time and position, and company features, namely company size and segment. The second part gathers information about the respondent's insight on employing KM in software testing, such as which software testing activity KM is more useful, test artifacts more appropriate for reuse, and what testing knowledge types are more important in software testing process. Questions were divided into multiple choice (with more than one option) and questions with answers that ranged from "very important" to "not important" on a scale based on the Likert Scale method [14]. The final average score represents the overall level of achievement or attitude toward a subject. Table 1 presents the Survey Questions (SQ)[1].

Table 1: Survey Questions

PART 1 - Participant's current profile
SQ1-SQ6. Name, Position (e.g., tester, test analyst, test designer, among others), Experience in this position (in years), Company Name (confidential), Company Size (micro, small, medium, large), Company Segment
PART 2 - Respondent's Perception
SQ7. In which software testing process activities does KM prove to be more useful?
SQ8. In which activities of testing planning does KM prove to be more useful?
SQ9. The test environment of a project is comprised of, among others, human, software and hardware resources that will be used to carry out the tests. Which of these features are most important to have available knowledge when defining the test environment?
SQ10. In which testing level do you consider KM more useful?
SQ11. Which knowledge type do you consider more important in the software testing process?
SQ12. Tacit knowledge can be made explicit, giving rise to explicit knowledge. In relation to the knowledge item types, indicate the degree of importance of generating explicit knowledge from tacit knowledge.
SQ13. With respect to test artifacts, indicate the degree of importance of knowledge reuse of these artifacts.
SQ14. Why KM is important for software testing (purposes)?

[1] *https://goo.gl/KHgzRK* - presents the complete questionnaire, with its questions and the answers format.

SQ15. What benefits does KM bring to software testing activities within your team?
SQ16. Would you consider applying KM in Software Engineering, especially in software testing process?

These questions were adapted from Souza et al. [7]. In [7], the questions were defined taking into account the results of a systematic mapping on KM in software testing [3] as well as in the conceptualization described by a software testing ontology, called ROoST: Reference Ontology on Software Testing [15].

The survey sample in [7] is a non-probabilistic sample, since the authors made the questionnaire available to groups of social networks interested in software testing and they also sent a direct e-mail to students, teachers and area researchers known to the authors. In total, 86 participants answered the survey [7]. The target audience in [7] did not provide a completely realistic view of the use of KM in software testing in practice in software development companies, since the participants are of different profiles, including, for example, undergraduate students. This fact became a motivation for our research. We applied the questionnaire in a specific target audience of professionals working with software testing in software development companies to gain their opinion into the application of KM in software testing.

Survey evaluation: before being made available, the questionnaire was validated by means of a pilot test. For this, a small sample was selected from the target audience to respond the survey. The questionnaire was also evaluated by an experienced researcher in Software Engineering and KM. The purpose of these assessments was to mitigate potential threats to validity (for example, identify possible difficulties that participants might have in understanding the questions) and correct any errors. After these assessments, the questions have been improved to make them clearer.

Survey application: The survey was made available directly to the 39 companies invited for a period of five months. In the end, 75 software testing professionals answered the questionnaire.

The last phase is the results analysis and presentation. This phase is presented in the next section.

3.1 Survey data analysis

As previously mentioned, 39 software companies participated in this research. Analyzing the first part of the questionnaire (*SQ1-SQ6*), the companies fell into the following categories: micro (9.33%), small (8%), medium (45.33%) and large (37.33%) companies. From the 39 companies, 75 employees answered the questionnaire, with an average of 2 respondents per company. Out of these 75 participants, 42.6% are testing analyst; 16% are testers; 12% are quality analysts; 8% work with agile development and apply Test Driven Development (TDD); and 21.3% perform other functions, such as quality assistant, test architect, test

coordinator, and test manager. On average the participants have experience in software testing of 5.8 years with a standard deviation of 11.46 years.

Concerning the second part of the questionnaire, Figure 1 presents a comparative between the results obtained in [7] and the results obtained in the survey presented here, regarding the most appropriate testing process activities to apply KM (*SQ7*). In this question, the following testing activities were considered: (i) Test Planning; (ii) Test Case Design; (iii) Test Code; (iv) Test Execution; and (v) Test Result Analysis. For software companies "Test Planning" (74.66%) was considered the testing activity in which KM is most useful. This perception is a few different from the survey conducted by Souza et al. [7], where "Test Case Design" (98.8%) and "Test Planning" (96.5%) were considered the most appropriate testing activities to apply KM. An explanation for this is that our survey was conducted only with professionals and 42.66% of the participants were test analysts. One of the main activities of test analysts is to plan how tests will be conducted. In order to have a more precise view of what testing activities are pointed out by participants from different positions, we present the graph in Figure 2. Note that most testing analysts point to Test Planning as most appropriate to apply KM.

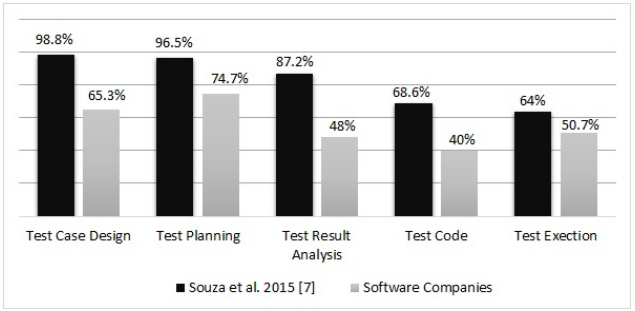


Fig. 1: Importance of KM to Software Testing Process Activities

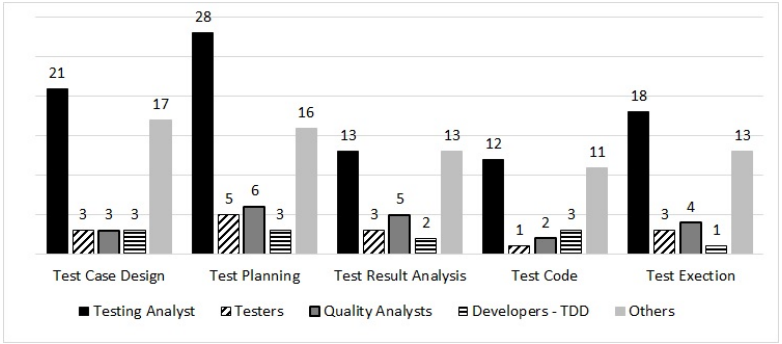


Fig. 2: Testing activities in relation to professionals

Regarding *SQ8*, we are interested in knowing in which activities of Test Planning does KM prove to be more important in the professionals insight, such as Testing Technique Selection, Test Environment Structuring, Test Plan Elaboration. As Figure 3 shows, "Test Plan Elaboration" (41.3%) was considered by companies that the testing planning sub-activity in which KM is most useful. Again, an explanation for this is that 42.66% of the participants were test analysts. On the other hand, in [7], "Testing Technique Selection" (41%) and "Test Environment Structuring" (36%) were considered the most appropriate testing planning sub-activities to apply KM.

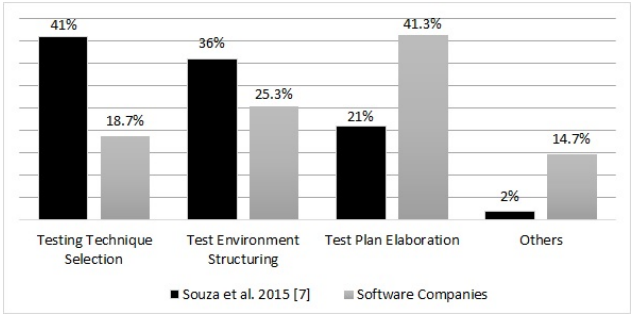


Fig. 3: Usefulness of KM in Testing Planning Sub-activities

The test environment considers both hardware and software, as well as the human resources involved in testing, that is, encompasses the entire structure where the test is performed [15]. With respect to *SQ9*, we asked about the most important resource to have available knowledge when defining the test environment. As Figure 4 shows, "Human resource" and "Software Resource" are considered the resources from which it is most important to have knowledge available at the time of setting the test environment in the perception of both the studies: software companies and Souza et al. [7]. However, there is a greater concern on the part of the companies in relation to software resources (50.6%). Based on these professionals' perspective, we analyzed the professionals profiles that chose Software Resource as being more important: 72% from testing analyst chose Software Resources; 66.6% from testers; 55.5% from quality analysts; 33.3% fom agile development; and 25% from others positions.

Software testing is usually carried out at different test levels [8]. The three most cited testing levels in the literature are made explicit, namely: *Unit Testing*, *Integration Testing* and *System Testing*. However, there may be others, such as *Regression Testing* and *Acceptance Testing*. In *SQ10*, we try to identify in which testing level KM is considered more useful. "System Testing" (73.3%) was considered the one in which KM more applies in relation the software testing professionals. This result is the same with the [7]. "System Testing" (48.8%) was also considered the testing level in which KM most applies.

Similarly to [7], *SQ11*, *SQ13*, and *SQ14* relate to types of knowledge items that are typically managed in the context of software testing and to the conver-

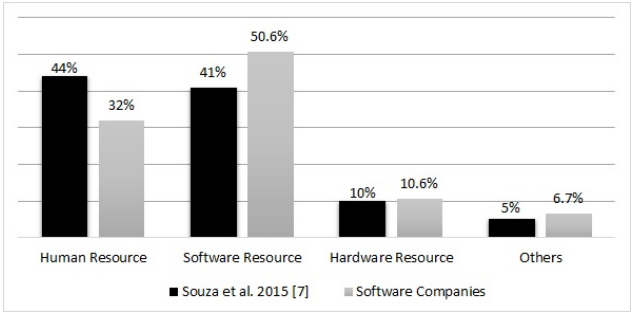


Fig. 4: Importance of KM to Testing Environment Definition.

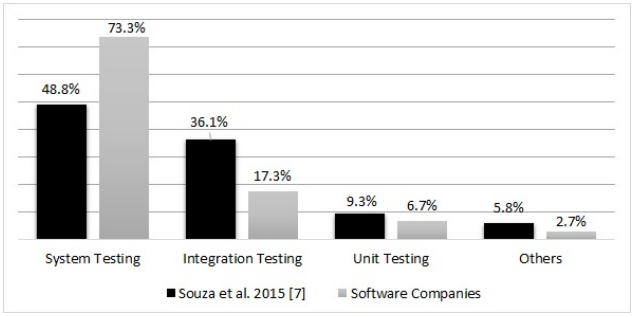


Fig. 5: Importance of KM to Test Levels.

sion of this knowledge. Souza et al. [7] reproduced the same aspects investigated in the systematic mapping about KM and software testing [3] to see if the survey participants confirm or deny the findings of the mapping. We have also conducted this analysis, however, as already mentioned, our target audience are just test professionals working in software development companies.

In *SQ11*, the aim is to highlight which knowledge type is most important to software companies. The participants considered explicit knowledge (74.66%) more important than tacit knowledge (25.33%). This is in line with the result obtained in [7]. However, according to one of the companies that participated in our research, the most important for a company is on how to transform the knowledge from tacit to explicit, that is, the explicit knowledge is more important, but as long as it is converted from tacit, so that it can be stored and reused. Another company also stressed that the conversion of tacit to explicit knowledge is more important, since it make the knowledge sharing easier. Furthermore, when new members are integrated into the team, learning becomes faster if knowledge is already stored.

Since the conversion of tacit knowledge to explicit is more important for companies, from *SQ12*, we try to identify which types of tacit knowledge items seem to be most important to be made explicit. As Figure 6 shows, “Communications

between the members of the test team" (73.33%) is the most important to be made explicit. On the other hand, in [7], in general, all items of knowledge were considered important to become explicit.

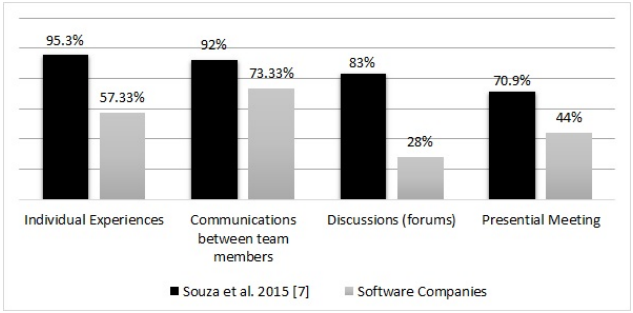


Fig. 6: Making tacit knowledge explicit.

Communication among team members, in particular, is strongly related to KM. Many companies that participated in the survey use agile methods. Agile methods prioritize interactions between people, promoting the face-to-face communication between team members [16]. In agile teams, the focus is the tacit knowledge and transforming this knowledge into explicit is considered a valuable asset for the organization. This statement corroborates the research results in questions *SQ11* and *SQ12*.

Regarding testing artifacts (*SQ13*), "Test Case" (70.66%) is considered the most reusable ones by the professionals, as shown in Figure 7. These results match what many researchers point out in test case reuse terms, as shown in systematic mapping conducted by [3]. [3] shows that the selection of regression test cases depends mainly on the experience of the testers and in order to provide a greater learning, it is important to have more detailed information on test cases.

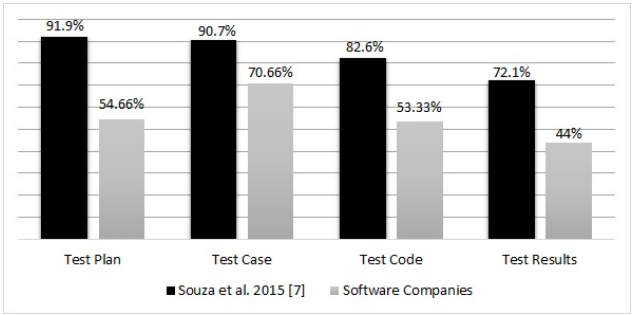


Fig. 7: Artifacts considered more appropriate for reuse.

Although in [7] the target audience is more generic, the results showed that "Test Plan" (91,9%) and "Test Case" (90,7%) are the most reusable artifacts. The authors also analyzed only the answers of professionals who work directly with software testing, and the result is that the most reusable artifacts are also "Test Cases" (94%).

In *SQ14* and *SQ15*, we intend to investigate how the purposes and benefits of applying KM in software testing were perceived as important by the software companies. For the companies, "Improving the quality of results" (68%) is the main purpose of applying KM in software testing, as presented in Figure 8. In [7], "Improving the quality of results" (28%) and also "Reducing costs, time and effort" (26%) were considered the main purposes.

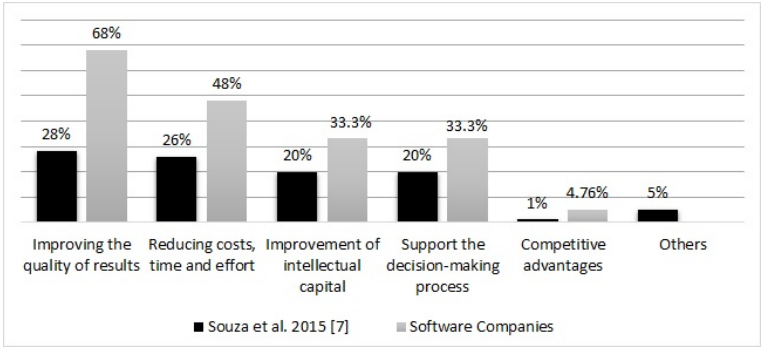


Fig. 8: Purpose of applying KM in Software Testing.

With respect to the benefits, "Increasing the testing process efficiency" (85.33%) and "Selecting and applying more suitable testing techniques" (61.33%) are considered by companies the main benefits of applying KM in software testing (see Figure 9). The same result was obtained in [7].

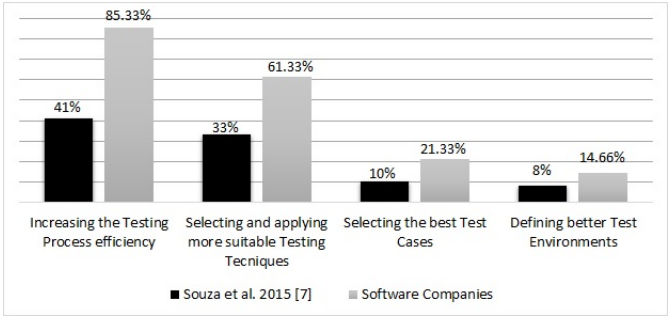


Fig. 9: Expected benefits of applying KM in Software Testing.

Applying KM in software companies is still a challenge, as there are still some barriers, such as KM systems are not appropriate yet, employees are normally reluctant to share their knowledge and increased workload [3]. In the survey question *SQ16*, we entirely leave to the participant's discretion to make any consideration about KM practices. Some comments corroborate exactly with the results of the systematic mapping conducted in [3] on KM in software testing.

According to the companies that participated in the research, applying 100% of the knowledge management process is not always possible. The employers need to meet aggressive deadlines and are pressed for results. This causes the team to be professionally overloaded, which leads to an emotional wear and dismay with the profession. Thus, processes that improve communication, reuse of information and working time become paramount for success.

Other comments emphasize that what is important is not only to implement KM practices, but also to be able to keep KM updated. The larger the company, greater is the challenge. The software testing area has gained prominence with the advent of several agile methods and frameworks. Agile practices can benefit organizations in the management and transfer of knowledge produced. According to participants, agile practices can become an important instrument to improve KM initiatives in software companies. However, companies still do not know how to reconcile these agile practices with KM practices to get better gains.

4 Related Work

As previously mentioned, the study conducted by Souza et al. [7], is a work directly related to ours, since we adapted and reapplied the survey conducted by Souza et al. [7]. However, we analyze only answers from software testing professionals working in Brazilian software development companies. A detailed results comparison of both studies is presented in the Section 3.1.

Tiwana [17] conducted a large-scale empirical study in 232 software development projects in 232 software companies. Three countries participated in the survey; Ireland, Russia, and India. The total number of responders was 54 from Ireland, 59 from Russia and 119 from India. As a result, Tiwana has identified that knowledge integration has a statistically significant effect on software development performance. With respect to software quality, knowledge integration lowers the density of warrant stage defects, lowers defect density throughout the development trajectory, improves software development efficiency, and increases design effectiveness. As in our case, this study conducted an empirical study in software companies in relation to the knowledge generated. However, our study focused only on Brazilian companies and in the context of software testing. Even so, regarding the findings on software testing and KM, both studies converge to the same perceptions about KM benefits.

In Aurum et al. [18], the current practice of KM in Software Engineering processes in two Australian companies is investigated. Similarly to our study, the authors also describes the KM activities and KM process used in Software Engineering practice. A total of 12 people were interviewed and answered a questionnaire in order to capture qualitative and quantitative data about aspects of

KM practices in four projects. One of the main findings in this study is that while KM was considered to be important, the tools, techniques and methodologies currently employed for software development were inadequate to address effective management of knowledge in these organizations; a uniform model of the KM process did not exist; and it is still necessary to encourage the sharing of tactical and explicit knowledge.

In a more recent study [19], Kuusinen et al. presents the results of a survey with 81 responses on organizational knowledge sharing in a multinational agile company. The survey concentrates on knowledge sharing practices, ease of knowledge sharing and motivation for knowledge sharing. Results show that using agile practices improves ease of knowledge sharing within teams. This confirms findings from our survey. In our survey, in *SQ16*, some participants mentioned the agile practices as facilitators for knowledge transfer. In fact, KM has been used to explain the distinctiveness of the agile approach.

5 Conclusions

In this paper, we reported on the results from an analysis related to the software testing professionals perception on the use of KM initiatives in software engineering companies. For this, we adapted and reapplied the survey conducted by Souza et al. [7]. From the survey results, we drew the following main conclusions: (i) Test Planning is considered the most important activity for using KM initiatives; (ii) Communications between the members of the test team is the type of tacit knowledge item most important to be made explicit; and (iii) Test Case is an artifacts that has received more attention by companies in terms of reuse. A strategic planning and the reusing existing test cases has a potential to significantly reduce software development costs and time.

One initial threat to internal validity in this work could be poor instrumentation, which could affect understanding the questions. For this reason, a pilot study was conducted as well as a survey validation by a KM and software testing researcher, so problems could be identified and corrected. Based on these threat, we intend to conduct interviews in software companies and compare the results. A second threat of this research concerns the sample size and representativeness. The sample had 75 valid answers among 39 software companies and a large concentration in Brazil's south and southeast regions, which affect both conclusion and external validity. The results cannot be generalized to a nationwide and world scale. Therefore, we intend to replicate this survey in other regions of Brazil and also in an international context.

A richer investigation with better mechanisms to perception about KM in software companies forms part of future work. We also intend to plan to complement our research with other empirical methods, for example, case studies based on real project data.

Acknowledgment

The authors would like to acknowledge the experts that agree to answer the survey.

References

1. Bjørnson, F.O., Dingsøyr, T.: Knowledge management in software engineering: A systematic review of studied concepts findings and research methods used. Information and Software Technology **11**(11) (2008) 1055–1068
2. Levy, M., Hazzan, O.: Knowledge management in practice: The case of agile software development. In: ICSE Workshop on Cooperative and Human Aspects on Software Engineering. (2009) 60–65
3. Souza, E.F., Falbo, R.A., Vijaykumar, N.L.: Knowledge management initiatives in software testing: A mapping study. Information and Software Technology **57** (2015) 378–391
4. Vasanthapriyan, S., Tian, J., Xiang, J.: A survey on knowledge management in software engineering. In: IEEE International Conference on Software Quality, Reliability and Security - Companion. (2015) 237–244
5. O'Leary, D., R., S.: Knowledge management: an interdisciplinary approach. Intell. Syst. **16**(1) (2001) 24–25
6. Itkonen, J., Mantyla, M., Lassenius, C.: The role of the tester's knowledge in exploratory software testing. IEEE Trans. Software Engineering (2013) 707–724
7. Souza, E.F., Falbo, R.A., Vijaykumar, N.L.: Using the findings of a mapping study to conduct a research project: a case in knowledge management in software testing. In: 41^{st} Euromicro Conference on Software Engineering and Advanced Applications. (2015) 208–215
8. Abran, A., Bourque, P., Dupuis, J.R., Moore, W.: Guide to the software engineering body of knowledge - swebok. Joint technical report, A project of the IEEE Computer Society Professional Practices Committee (2004)
9. Spender, J.C.: Making knowledge the basis of a dynamic theory of the ïňĄrm. Strategic Management Journal **17** (1996) 45–62
10. Markkula, M.: Knowledge management in software engineering projects. In: 11^{th} International Conference on Software Engineering and Knowledge Engineering. (1999) 20–27
11. Nonaka, I., Takeuchi, H.: The knowledge-creating company. 1 edn. Oxford University Press, Oxford, USA (1997)
12. Ciolkowski, M., Laitenberger, O., Vegas, S., Biffl, S.: Practical experiences in the design and conduct of surveys in empirical software engineering. Lecture Notes in Computer Science **2765** (2003) 104–128
13. Kitchenham, B., P.S.L.: Guide to Advanced Empirical Software Engineering. Personal Opinion Surveys. Chapter 3. Springer-Verlag London (2008)
14. Likert, R.: A technique for the measurement of attitudes. Archives of Psychologyy **22**(140) (2010) 1–55
15. Souza, E.F., Falbo, R.A., Vijaykumar, N.L.: ROoST: Reference Ontology on Software Testing. Applied Ontology **12**(1) (2017) 59–90
16. Crispin, L., Gregory, J.: Agile testing:a practical guide for testers and agile teams. Pearson Education, Boston, MA (2009)
17. Tiwana, A.: An empirical study of the effect of knowledge integration on software development performance. Information and Software Technology (2004)
18. Aurum, A., Daneshgar, F., Ward, J.: Investigating Knowledge Management practices in software development organisations - An Australian experience. Information and Software Technology **50** (2008) 511–533
19. Kuusinen, K., Gregory, P., Sharp, H., Barroca, H., Taylor, K., Wood, L.: Knowledge sharing in a large agile organisation: A survey study. In: International Conference on Agile Software Development. (2017) 135–150

View publication stats