Investigating the Developer's Perceptions of Unit Testing and Its Practice

Lidvaldo Santos[1], **Fernanda Santos**[2], **Renan Parreira**[3], **Rafael de Mello**[4]

Programa de Pós-Graduação em Informática - PPGI
Universidade Federal do Rio de Janeiro (UFRJ)

{lidiosantos,renan}@ufrj.br,fernandagarciasantos@gmail.com

rafaelmello@ic.ufrj.br

Abstract. *Unit testing is an important and largely disseminated software engineering activity, contributing to the prevention of defects in source code elements before running more comprehensive tests. However, developers typically perform unit tests ad-hoc, lacking in coverage and quality. Consequently, the unit testing practice remains informal and strongly influenced by developers' initiative. We report in this paper an opinion survey on the developers' perceptions of unit testing and its practice. We conducted the survey with 32 developers from two Brazilian companies. The survey findings revealed that developers from both companies tend to share positive beliefs about the relevance of unit testing. However, they also tend to share negative perceptions about its practice in their working environments. Besides, we found a positive moderate correlation between the developers' motivation and their perceptions about the training, frequency, and time allocated for unit testing. The survey results reveal the dissonance between the developers' beliefs about unit testing and their practice, influenced by issues in their working environments. This dissonance may lead to a lack of commitment and internal conflicts.*

1. Introduction

Development teams should continuously perform software testing activities to verify the quality of the software systems being implemented and evolved. Software testing may be performed at different levels, addressing different development stages and requiring the involvement of individuals playing different roles [Garousi and Mäntylä 2016]. In particular, unit testing has been widely recognized as a valuable software testing practice for improving software reliability, allowing the detection of flaws, errors, and defects at the early stages of coding activities [Xie et al. 2016]. [Prado and Vincenzi 2018] define unit testing as code-level testing, typically conceived and performed by the same software developers that authored the source code. Each unit test should address a single lowest cohesive code element (unit).

According to [Martins and Braga 2016], the unit testing practice contributes to the software development process in several ways, including the reduction of defects, the improvement of the source code documentation, the reduction of concerns with code changes, and the protection of the source code against undesired and accidental modifications. In this way, we may consider unit testing an essential software verification activity supported by many automated tools [Virgínio and Machado 2021]. However, these technologies seem to have not been sufficient to promote the practice of unit testing in a planned, comprehensive, and systematic way. According to [Prado and Vincenzi 2018], there is no theory on how professionals perform unit testing activities. Besides, the support that current automated testing tools offer is often limited to eliminating repetitive manual efforts, such as assertion verifications, test data generation, and coverage checking. Although unit testing is one of the most widely used practices to find bugs in software systems, there is a lack of cognitive support for unit testing [Prado and Vincenzi 2018], which may discourage the practice [Wang and Offutt 2009]. Unit testing is frequently considered a tedious task that developers have little motivation to perform [Saloum 2019], which can reduce the quality and the coverage of the unit tests performed. Despite the several tools available for automated testing [Rana et al. 2019], several developers are few or not motivated to write unit tests for their source code, lacking in performing the tasks [Lappalainen et al. 2010, Wang and Offutt 2009].

We understand that conceiving effective technologies for stimulating unit testing practice in development teams requires understanding the developers' beliefs about unit testing and the main challenges faced in performing this task in their working environments. In this paper, we report the first execution of an opinion survey on the developers' perceptions of unit testing and its practice in their working environments. We count on the participation of 32 developers from two companies. The study findings reveal dissonance between the developers' positive beliefs about unit testing and the limited practice, probably promoted by issues faced in their working environments. We also found that the developers' perceptions about frequency, training, and time allocated are moderately correlated with their motivation for writing unit tests. This is the first step of our ongoing research on the sociotechnical aspects of the unit testing practice. Through our research, we expect to build evidence-based technologies to enhance and stimulate the unit testing practice.

2. Study Design

We report in this paper an empirical study we conducted to characterize the perceptions about unit testing and its practice in Brazilian software development teams. The protocol applied in the study follows predominantly qualitative and exploratory perspectives. For this study, we established the following research questions:

RQ1) What are the developers' perceptions about unit testing?

RQ2) What are the developers' perceptions about the unit testing practice in their working environments?

To answer RQ1 and RQ2, we opted for conducting an opinion survey [Linaker et al. 2015]. Opinion surveys are largely conducted in the field to characterize the state of practice by gathering practitioners' perceptions about software engineering activities [de Mello et al. 2014b, de O. Passos et al. 2018] and the daily work [Li et al. 2008, de Souza et al. 2023]).

2.1. Talks with Developers

We may characterize the developers' perceptions of unit testing and its practice from several perspectives. However, we understand some of them should be more relevant for the practice based on our academic research. Besides, we should use appropriate and simple language when designing survey instruments, avoiding too complex questions [Linaker et al. 2015, Torchiano et al. 2017].

To support the design of our survey, we conducted informal talks with five software developers working at a private bank in Brazil. Along these talks, we stimulated developers to provide information about their professional experience and development activities. More specifically, we asked developers to describe their perceptions of unit testing practice, including its frequency, the resources employed, and how they evaluate the quality and coverage of unit tests developed. Besides, we asked about their motivation for performing unit testing and the main challenges faced performing the activity.

The five developers signaled that they often run unit tests to evaluate the main features of their software systems. Besides, most of them understand that unit testing plays an important role in promoting the quality of software systems once it allows developers to identify problems in the source code. Moreover, they argued that unit testing leads developers to understand the application behavior.

Regarding the existing motivation for unit testing, most developers commented about the role of unit testing in promoting team collaboration. In this sense, they understand that unit testing is useful for supporting teams in restoring an existing baseline. About time allocated to unit testing, most respondents understand it would be sufficient.

Some developers pointed out the lack of experience in continuously integrating unit testing. Besides, they claimed a lack of proper training in unit testing. For instance, P4 stated: ”I had no training. I learned to do tests on my own and with the support of my colleagues. I had only two months to adapt it!” In his turn, P5 argued that the company should invest in knowledge and training for the team. Despite that, P2 understands that ”the company encourages developers to carry out unit testing, which is usually a standard in development teams.”

2.2. The Survey Questionnaire

Based on the issues raised in talks with developers, we could design an optimized survey questionnaire published in Google Forms[1]. After the agreement with the consent form, the participant should answer the following characterization questions:

Q1. What is your education level? (closed question, nomimal)

Q2. How long have you been working with software development? (closed question, in years)

Q3. What activities do you perform in your work environment? (open question)

With Q3, we want to characterize the main roles played by the survey respondents. Our experience shows that using an open question to ask about the developers' activities is a useful strategy in surveys to bring out context to the developers' individual experience [de Mello et al. 2014a, de O. Passos et al. 2018]. After answering the characterization questions, the participants should answer nine specific questions on their perceptions of unit testing and its practice. These questions were designed as statements to be assessed through the same 5-level Likert scale of agreement [Allen and Seaman 2007].

This scale offers as a response option a set of points with descriptions that indicate

[1] https://tinyurl.com/4693zt9c

extremes such as ”Strongly Agree” and ”Strongly Disagree”, which allows the intensity levels of the opinion to be identified.

For each statement, the participants should indicate an option based on their personal experience with unit testing in their current working environment:

Q1. I consider the development of software unit tests relevant.

Q2. I frequently perform unit testing in software development.

Q3. I note that the software development team has received training to carry out tests.

Q4. I consider the time allocated to unit testing sufficient.

Q5. It is essential to submit unit tests for continuous integration.

Q6. I consider test coverage relevant in software development.

Q7. It is important to send unit tests for code review.

Q8. While running unit tests, I can spot flaws in my own code.

Q9. During software development, I feel motivated when writing unit tests.

2.3. Population and Sampling

The survey’s target audience is software developers who work with unit testing. Considering the purpose of the survey, we understand that we should listen to developers with different levels of experience with unit testing. The unit of analysis in this study is the software developer. However, we understand it is important to analyze the influence of the developers’ working environment over the unit testing practice. Thus, we opted to establish a convenience sample composed of software developers from two IT consulting companies (Company A and Company B).

Company A has been in the market for over 30 years, with several projects across Brazil and the United States. With more than 3,000 employees, Company A has a well-defined development process based on Scrum for new projects and Kanban practices for maintenance with activity management and delivery measurement indicators. The developers recruited from Company A work on a project for providing software solutions to a client from the oil and gas distribution sector. The project activities include developing and maintaining core software systems for supporting the business management of this client. Although Company A teams follow a formal process for systems development and maintenance, this process does not cover steps, roles, and resources for software testing. Besides, developers from Company A typically run unit testing ad-hoc without automated support.

Company B is a multinational created 23 years ago by the merger of two IT giants. Currently, Company B has a staff of around 30,000 employees worldwide. The company has a well-defined development process based on Scrum and Kanban, managing the development activities and deliveries with the support of the Azure tool. The same development process is followed in all projects. After the end of each development task, the technical leader verifies the number of unit testing performed through Bitbucket. We recruit developers from Company B working on different projects at the Brazilian headquarters.

The sampling strategy followed to recruit developers from these companies was convenience. The survey invitation was sent by e-mail to each developer.

2.4. Characterization of Participants

The characterization of the survey participants revealed that samples from both s companies are diverse in their main roles. In Company A, respondents are distributed between front-end developers, back-end developers, and software architects. In company B, we found front-end developers, back-end developers, full-stack developers, and technical leaders. Participants from both companies carry on activities related to developing new software applications and maintenance. Concerning professional background, 20% of the Company A respondents declared having experience of up to five years with software development, while 20% have experience of up to ten years in the field. The other 60% have worked in the field for more than ten years. In turn, 31% of respondents from Company B have experience of up to five years in software development, 13% have experience of up to ten years, and the other 66% with software development for more than ten years. Furthermore, all respondents from Company A and most of the Company B respondents declared experience with Java technology for developing web applications.

The characterization of the survey participants indicates that most respondents from both companies have extensive experience in software development, predominantly working with the production of source code on the Web. Besides, the distribution of higher education levels is quite similar, with few exceptions. In the next subsections, we present and discuss the developer's answers to the specific survey questions, by company.

3. Results and Analysis

The survey was carried out at the end of 2022, counting the participation of 32 developers, 10 from Company A and 22 from Company B. We report in the following subsections the survey results.

3.1. Company A

Figure 1 and Figure 2 present the distribution of the answers given by the ten respondents from Company A for the specific questions. Figure 1 addresses the four survey questions related to the developers' perceptions about the relevance of unit testing practice. Based on the answers presented, one may see the respondents completely or partially agree about the relevance of developing unit tests and assuring their coverage. This positive perception of the relevance of unit testing is reflected in the agreement of most developers that unit tests should be reviewed and continuously integrated. Therefore, these results indicate that developers from Company A tend to believe that unit testing should not be an occasional or ad-hoc practice, requiring its proper insertion in the software development process. Figure 2 presents the distribution of the developers' answers to questions about their practice and motivation for unit testing. Analyzing the developers' perceptions, we identified some possible aspects of their working environment that may negatively affect the performance and coverage of unit tests. According to the data collected in Company A, the respondents tend to disagree that the team receives training on unit testing. Besides, developers tend to disagree that the time allocated for unit testing is sufficient.

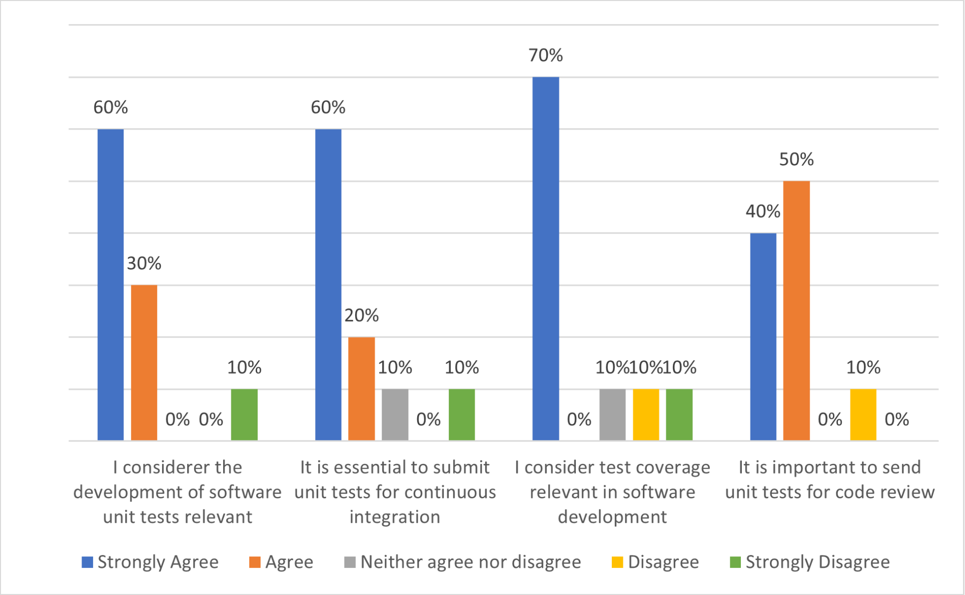

Figure 1. Developers' perceptions about unit testing (Company A).

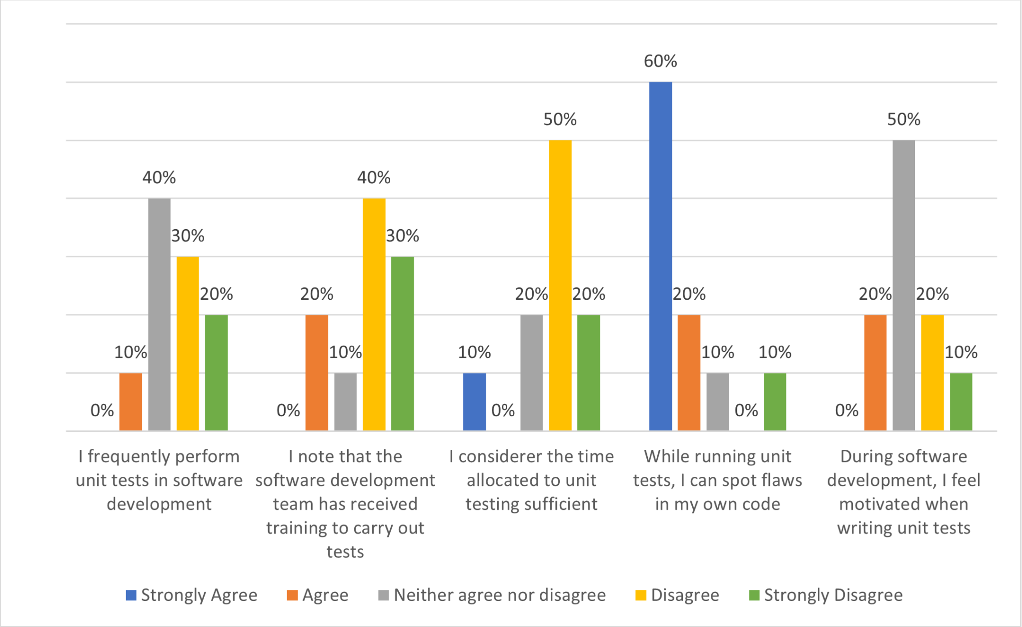

Figure 2. Perceptions about unit testing practice (Company B).

These aspects possibly hamper the motivation of the developers for writing unit tests even though they tend to experience an effective contribution of unit testing to the process, i.e., spotting flaws in the source code. Consequently, one may see that most developers tend to recognize that they do not perform unit tests frequently. The predominantly negative results about the unit testing practice suggest that Company A lacks in encouraging and supporting developers to unit testing despite the developers recognizing the relevance of this activity. Besides, the low frequency of performing unit tests suggests that developers are not conducting comprehensive unit tests to ensure the quality of the developed software systems.

3.2. Company B

Figure 3 and Figure 4 present the distribution of the answers given by the 22 respondents from Company B for the specific questions. Figure 3 addresses the four survey questions related to the developers' perceptions about the relevance of unit testing practice. Based on the answers given, we found that most respondents fully or partially agree on the relevance of unit testing and assuring its coverage. Besides, the respondents tend to agree that unit tests should be submitted for continuous integration and code review. Therefore, these results indicate that developers from Company B tend to believe that unit testing should not be an occasional or ad-hoc practice, requiring its proper insertion in the software development process.

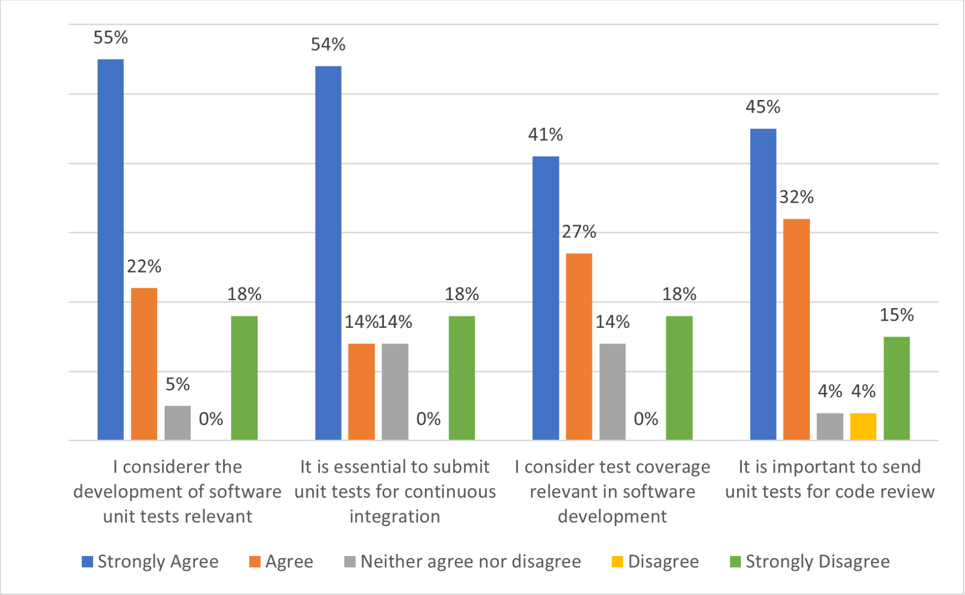

Figure 3. Developers' perceptions about unit testing (Company B).

Figure 4 presents the distribution of the developers' answers to the questions addressing their practice and motivation for unit testing. The Company B developers' answers reveal a lack of trend in their perception of specific training given by the company for unit testing and their perception of the proper time allocated for performing this task. The feeling of motivation for writing tests also divides the developers' opinions despite most respondents agreeing that unit testing allows them to spot flaws in their source code. Besides, more than half of the respondents agree that they frequently perform unit tests during their development activities. The results suggest that the practice of unit testing in Company B is not sufficiently supported despite being promoted in the development process through coverage goals (see Section 2).

4. Discussion

We found that developers from both companies tend to have positive perceptions about unit testing (RQ1). Most respondents consider the development of unit tests relevant is important to assure its coverage, review, and continuous integration. These findings reveal the need to insert unit testing into the software development process.

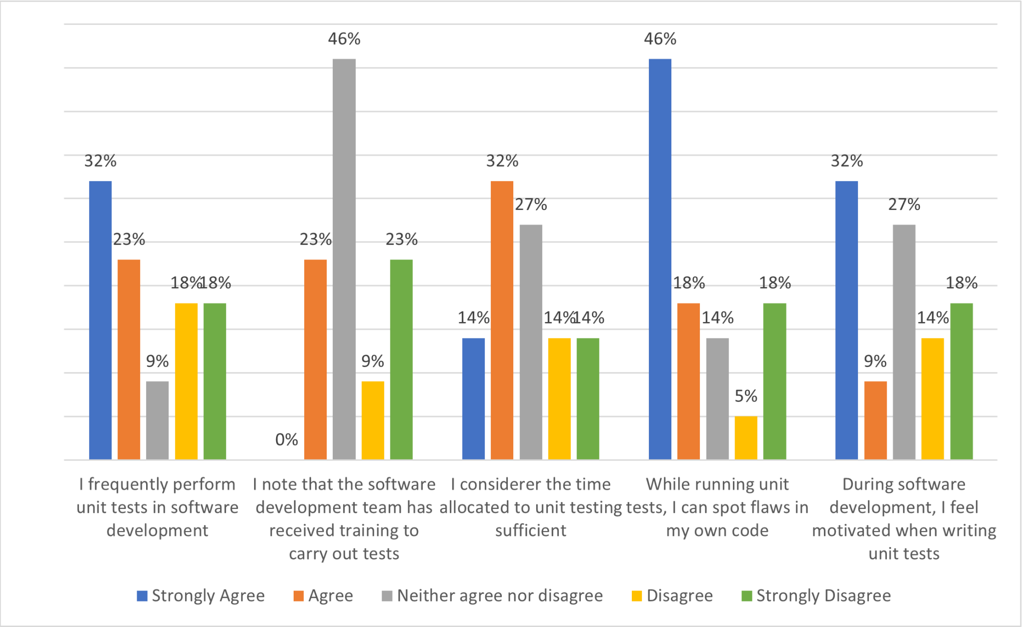

Figure 4. Perceptions about unit testing practice (Company B).

The similarities of the developers' beliefs from both companies about unit testing combined with their background suggest that the samples investigated would be able and sufficiently motivated to perform unit tests if properly supported by their working environments. However, when asked if they agree that unit tests are frequently performed during development, we found divergent results between Company A and Company B (RQ2). Besides, we found that several developers from both companies tend not to feel motivated to write unit tests.

Thus, considering that developers from both companies tend to agree about the relevance of unit testing for the software development process, we should analyze the possible environmental factors that may harm their motivation and frequency. The higher motivation observed among developers from Company B may be partially explained by the much more positive perception of time allocation and training among Company B developers in comparison with the developers from Company A. Only 10% of the respondents tend to agree that the time allocated to unit testing is sufficient while almost half of the respondents tend to agree that the time to carry out tests is sufficient. Besides, 70% of Company A developers tend to negatively perceive the offering of training in unit testing, while this negative perception is shared only by 28% of developers from Company B. These findings point to challenges partially addressed in [Runeson 2006] survey findings. In this investigation, the author identified that the main difficulties for unit testing are related to the lack of organizational support, leading developers to perform unit testing predominantly ad-hoc and when possible.

Considering the diverse distribution of motivation for unit testing in the whole sample (32 developers), we also analyzed possible correlations between the developers' perceptions about the unit testing practice in their teams and their motivation. We hypothesize that more positive perceptions about the working environment (frequency, training, time) for unit testing are positively correlated with the higher motivation of developers in writing unit tests. For this purpose, we applied the paired Spearman correlation test. The results revealed that the developers' perceptions about *frequency*, *training*, and *time allocated* are moderately correlated with the developers' motivation for writing unit testing (0.4593, p-value=0.0081; 0.5443, p-value=0.0016; 0.5548, p-value=0.0010).

The correlations found indicate that the developers' perceptions about unit testing practice (RQ2) influence their motivation when unit testing, although it is insufficient to establish a causal relationship. Consequently, this finding suggests that being aware of the relevance of unit testing (RQ1) is insufficient for stimulating the practice without a proper working environment.

5. Conclusion

In this paper, we report a survey on the developers' perceptions about unit testing and its practice. As far as we are aware, this is the first empirical study comparing developers' perceptions of unit testing and its practice in distinct working environments. The generalization of the findings reported in this study may be considered limited. However, we understand that the comparisons made between the perceptions of developers from the software companies reveal common issues in their working environments that may also address other companies. Besides, we intend to improve the research instrument in future replications, adopting open-ended questions to gather the developers' arguments for their answers.

The survey findings indicate that we should go in-depth on characterizing developers' beliefs about unit testing and their practice considering their working environments. Despite the limitations of the survey instrument, we could observe a concerning dissonance between the developers' beliefs and their practice. According to Social Psychology [Hogg and Vaughan 1995], such dissonance is a potential trigger for lack of commitment and internal conflicts. In this way, we intend to conduct an observational study with a development team, including the planning, execution, and verification of unit testing during the development process. In this study, we also plan to investigate the developers' social representations [Moscovici 1988] of unit testing, aiming at revealing possible hidden team-level obstacles and challenges for improving the unit testing practice.

References

Allen, I. E. and Seaman, C. A. (2007). Likert scales and data analyses. *Quality progress*, 40(7):64–65.

de Mello, R., da Silva, P., and Travassos, G. (2014a). Agilidade em processos de software: Evidências sobre características de agilidade e práticas Ágeis. In *Anais do XIII Simpósio Brasileiro de Qualidade de Software*, pages 151–165, Porto Alegre, RS, Brasil. SBC.

de Mello, R. M., da Silva, P. C., and Travassos, G. H. (2014b). Agilidade em processos de software: Evidências sobre características de agilidade e práticas ágeis. In *Anais do XIII Simpósio Brasileiro de Qualidade de Software*, pages 151–165. SBC.

de O. Passos, A. F., de Freitas Farias, M. A., de Mendonça Neto, M. G., and Spínola, R. O. (2018). A study on identification of documentation and requirement technical debt through code comment analysis. In *Proceedings of the XVII Brazilian Symposium on Software Quality*, pages 21–30.

de Souza, A., dos Santos, F., dos Santos, L., Belém, P., da Silva, S., Vivacqua, A., and de Mello, R. (2023). Investigando a percepção de feedback em times de desenvolvimento de software: Um estudo inicial. In *Anais do VIII Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software*, pages 91–100. SBC.

Garousi, V. and Mäntylä, M. V. (2016). A systematic literature review of literature reviews in software testing. *Information and Software Technology*, 80:195–216.

Hogg, M. A. and Vaughan, G. M. (1995). *Social psychology: An introduction.* Harvester Wheatsheaf.

Lappalainen, V., Itkonen, J., Isomöttönen, V., and Kollanus, S. (2010). Comtest: a tool to impart tdd and unit testing to introductory level programming. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, pages 63–67.

Li, J., Conradi, R., Slyngstad, O. P., Torchiano, M., Morisio, M., and Bunse, C. (2008). A state-of-the-practice survey of risk management in development with off-the-shelf software components. *IEEE Transactions on software Engineering*, 34(2):271–286.

Linaker, J., Sulaman, S. M., Höst, M., and de Mello, R. M. (2015). Guidelines for conducting surveys in software engineering v. 1.1. *Lund University*, 50.

Martins, T. F. R. and Braga, T. F. (2016). Framework para geração de teste unitário.

Moscovici, S. (1988). Notes towards a description of social representations. *European journal of social psychology*, 18(3):211–250.

Prado, M. P. and Vincenzi, A. M. R. (2018). Towards cognitive support for unit testing: A qualitative study with practitioners. *Journal of Systems and Software*, 141:66–84.

Rana, I., Goswami, P., and Maheshwari, H. (2019). A review of tools and techniques used in software testing. *Int. J. Emerg. Technol. Innovative Res*, 6(4):262–266.

Runeson, P. (2006). A survey of unit testing practices. IEEE software, v. 23, n. 4, p. 22-29.

Saloum, R. (2019). The impact of gamification in unit testing. Department of Computer Science and Engineering - University of Gothenburg.

Torchiano, M., Fernández, D. M., Travassos, G. H., and de Mello, R. M. (2017). Lessons learnt in conducting survey research. In *2017 IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)*, pages 33–39. IEEE.

Virgínio, T. G. A. and Machado, I. (2021). Avaliação empírica da geração automatizada de testes de software sob a perspectiva de test smells. In *Anais Estendidos do XII Congresso Brasileiro de Software: Teoria e Prática*, pages 112–126. SBC.

Wang, S. and Offutt, J. (2009). Comparison of unit-level automated test generation tools. In *2009 International Conference on Software Testing, Verification, and Validation Workshops*, pages 210–219. IEEE.

Xie, T., Tillmann, N., and Lakshman, P. (2016). Advances in unit testing: theory and practice. In *Proceedings of the 38th international conference on software engineering companion*, pages 904–905.