Perception of Professionals Regarding Behavior-Driven Development (BDD): A Descriptive and Statistical Study

Shexmo Richarlison Ribeiro dos Santos[a], Gustavo S. Melo[b], Michel S. Soares[c] and Fabio Gomes Rocha[d]
Federal University of Sergipe, Department of Computing, São Cristóvão, Brazil

Keywords: Behavior-Driven Development (BDD), Framework, Software Architecture, Software Engineering, Survey.

Abstract: **Context:** Analyze the perception of professionals who use the Behavior-Driven Development (BDD) framework in software development. **Problem:** Identify the aspects inherent to adopting BDD by the software industry. **Solution:** Through descriptive and statistical analysis, advance understanding of the characteristics of using BDD. **Method:** Carry out a Survey to characterize BDD, where the target audience is professionals who use this framework in their work activities. **Summary of results**: The Survey carried out in this study was answered by 43 professionals to characterize how BDD has been adopted in software development. **Contributions and impact**: The main contribution of this article is that the lack of experience in using BDD directly impacts the performance of work activities. Thus, it is necessary to have experience in adopting BDD to achieve the potential expected by this framework.

1 INTRODUCTION

Proposed in 2003 by Dan North (North, 2006), Behavior-Driven Development (BDD) is a framework used throughout the software life cycle, mainly in the requirements elicitation and automation testing stages (Moe, 2019), (Silva and Fitzgerald, 2021). BDD was created to mitigate the problems that arise from test-driven development (TDD). While TDD focuses on unit testing, BDD has the behavior that the system will perform as its objective upon a given action.

Following part of the *Goal-Question-Metric* model proposed by Mashiko and Basili (Mashiko and Basili, 1997), the main goal of this study is to "**Analyze** aspects of BDD, **with the purpose of** characterizing, **concerning** the use of this framework, **from the point of view** of professionals, **in the context of the** software life cycle". In this way, by characterizing the aspects inherent in adopting BDD, it will be possible to embrace it more effectively.

The presented survey is justified by the growing use of BDD in software development (Ribeiro dos Santos et al., 2024), causing the need for its users' perspective on the framework. Through this study, it will be possible to advance understanding regarding adopting BDD through professionals who use it, aiming to present the state of practice of this framework.

2 BEHAVIOR-DRIVEN DEVELOPMENT (BDD)

With the aim of faster and more assertive deliveries, Behavior-Driven Development (BDD) was created by Dan North (North, 2006) to mitigate the problems caused by Test-Driven Development (TDD), for example, rework caused by the elicitation of requirements that do not correspond to the behavior expected by software. In this way, BDD has a methodology focused on software behavior to pay more attention to how requirements are elicited, both functional and non-functional, e.g., software quality (Soares and Vrancken, 2011).

BDD adopts ubiquitous language called *Gherkin*. *Gherkin* is a fluid daily language whose primary goal is to be understood by the interested parties in the process. Therefore, it has greater clarity in eliciting requirements and improved communication (Pereira et al., 2018) (Santos et al., 2024), contributing to both the elicitation of functional and non-functional

[a] https://orcid.org/0000-0003-0287-8055
[b] https://orcid.org/0009-0007-8141-2064
[c] https://orcid.org/0000-0002-7193-5087
[d] https://orcid.org/0000-0002-0512-5406

Santos, S. R. R., Melo, G. S., Soares, M. S. and Rocha, F. G.
Perception of Professionals Regarding Behavior-Driven Development (BDD): A Descriptive and Statistical Study.
DOI: 10.5220/0013355600003929
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 264-271
ISBN: 978-989-758-749-8; ISSN: 2184-4992
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

requirements to enhance the quality of the software architecture.

The presented fields are always filled in with user stories and then automated through testing. The delivery of the final product becomes more assertive when it is commonly discussed and understood among interested parties.

There is an initial meeting called “3 amigos” where the product owner, the tester, and the developer meet to elicit the requirements more precisely and more assertively regarding what is expected about the product behavior of the elicited features. From the requirements elicitation, the developer and the tester carry out the automation and validation of the functionalities requested by the product owner, respectively, to deliver small parts of the software throughout the process to put it into operation in the shortest possible time. If the desired behavior is achieved, proceed to the requirement documentation. Otherwise, it is necessary to return to the requirement elicitation stage to readjust what is expected from the software. Thus, requirements are verified and validated to identify whether what was requested was done and implemented (Bruschi et al.,).

Functionalities written using BDD are straightforward so that everyone involved in the project understands them. BDD is used throughout the software life cycle to help with requirements elicitation, validation and documentation, and software maintenance. Using the “given, when, then” pattern, BDD can outline the requirements more objectively, generating a reduction in rework due to poorly planned elicitations.

3 SURVEY

A study focused on BDD was conducted, where a survey was applied with the aim of better understanding its practice from the point of view of professionals in the job market. According to Shaw (Shaw, 2003), procedures or techniques are new or better ways to perform tasks in Software Engineering. Identifying aspects related to BDD becomes essential to understand the framework, which, based on such results, can help professionals and researchers find a better way to use and understand it for new research, respectively.

3.1 Research Goal

The objective of this study follows part of the *Goal-Question-Metric* (Mashiko and Basili, 1997) model, namely: “**Analyze** aspects of BDD, **with the purpose of** characterizing, **concerning** the use of this framework, **from the point of view** of professionals, **in the context of the** software life cycle”. A survey consisting of 12 closed and mandatory questions was created in Table 1 to achieve the proposed goal. Among questions Q2 to Q5, there was an open space if the participants considered it necessary to complement their responses.

When creating the questions for this research, the population was defined as professionals who use BDD in their work activities (Kitchenham and Pfleeger, 2002).

3.2 Pilot and Execution

A pilot was carried out with three respondents, and minor adjustments were made regarding the formatting and clarity of some questions, which were incorporated into the published version of the survey.

Initially, the research was released, in its English and Portuguese versions, only to companies through *Whatsapp* groups and the *LinkedIn* platform, and, later, the *snowball* method was used to get more respondents. The *snowball* method asks respondents to send the questionnaire to acquaintances who may be part of the research’s target audience.

4 RESULTS

The survey was open for responses for a month, and 43 respondents were reached. The answers to the research questions in Table 1 are as follows.

Q1 - How Long Have You Used the BDD Framework? According to Fig. 1, a decreasing order can be observed concerning the time of use of BDD and the number of professionals who use it, so that the longer the experience, the fewer respondents one can obtain.

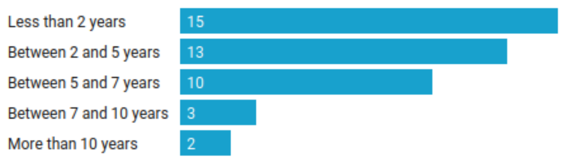

Figure 1: Time of use of BDD.

Interestingly, this gradual increase can be seen among the respondents, so it can be inferred that BDD has become more popular for software development in recent years. The increasing aspect is also mentioned by Santos, Rodriguez, and Rocha (Ribeiro dos Santos et al., 2024).

Table 1: Survey questions.

ID	Question	Justification
Q1	How long have you used the (BDD) framework?	Understand the professional's experience.
Q2	What is the main tool you use for BDD?	Identify the most used tools.
Q3	Where do you use BDD most?	Explain in which part of the process BDD has been most applied.
Q4	What are the main benefits related to the adoption of BDD?	Elicit the main positive characteristics of BDD.
Q5	What are the main difficulties encountered for the adoption of BDD?	Identify whether there are negative points inherent to adopting BDD.
Q6	On a scale of 0 to 10, how much is (*Gherkin*) understandable?	Identify the effectiveness of the *Gherkin* language.
Q7	On a scale of 0 to 10, considering the purpose of the BDD regarding readability, how much does BDD achieve in this regard?	Check readability related to the use of BDD.
Q8	On a scale of 0 to 10, considering the purpose of the BDD regarding communication, how much does BDD achieve in this regard?	Check communicability related to the use of BDD.
Q9	On a scale of 0 to 10, how much does BDD perform faster delivery?	Realize the efficiency in using BDD.
Q10	On a scale of 0 to 10, how much does BDD perform to deliver with higher quality?	Understand the effectiveness of using BDD.
Q11	On a scale of 0 to 10, how practical is BDD to perform updates in the code?	Explain the practicality of BDD concerning living documentation.
Q12	On a scale of 0 to 10, how much does BDD contribute to the relationship to the company's economy when having living documentation?	Check whether living documentation can generate financial savings for the company.

Q2 - What Is the Main Tool You Use for BDD? According to Fig. 2, 12 tools were identified, so the four most cited ones can be highlighted, with Cucumber being the most used, where it is worth emphasizing that it works in several languages, such as Java, for example; Behavior, a framework for the Python language; Specflow, which is a framework for .Net; and JBehaviour, which is the Java framework created by Dan North.

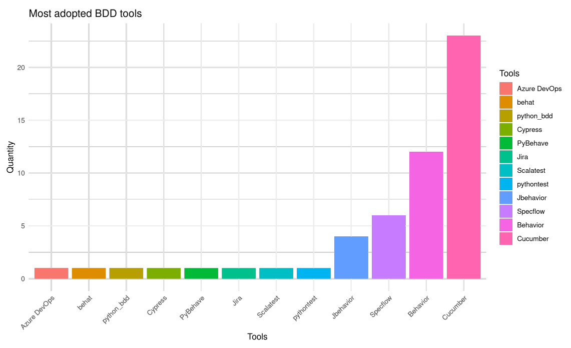

Figure 2: Most used tools when using BDD.

Through this graph, it is possible to understand which tools BDD is most used for to infer which ones can work more effectively and more straightforwardly.

Q3 - Where Do You Use BDD Most? According to Fig. 3, one can observe the frequency of BDD use in the software development process stages to highlight the test stages, requirements, and test cycles with the highest number of mentions.

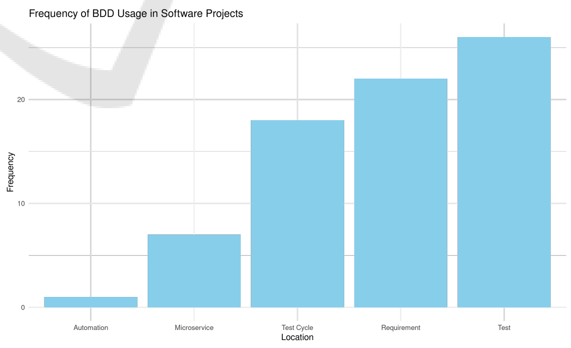

Figure 3: Places where BDD is most used.

As BDD is a framework used throughout the software life cycle, it was also mentioned in other stages but on a smaller scale. Furthermore, it is interesting to highlight the following response from one of the participants:

Despite being aware of the ability to automate tests with BDD, such as Cucumber, I use BDD *to specify requirements, detail the user story narrative, and identify user acceptance criteria that can be used as scenarios or test cases, patterns, and exceptions.*

The use of BDD can be seen in terms of its readability to help professionals when dealing with user story narratives and their respective acceptance criteria. Through this quotation, using BDD demonstrates as a positive point the use of the *gherkin* language and the "given, when, then" pattern to be used in a simple way to achieve behaviors desired, the primary purpose of Dan North (North, 2006).

Q4 - What Are the Main Benefits Related to the Adoption of BDD? According to Figure 4, it is possible to observe the main benefits according to the respondents, highlighting communication as the main benefit, followed by traceability, requirements, behavior, and readability.

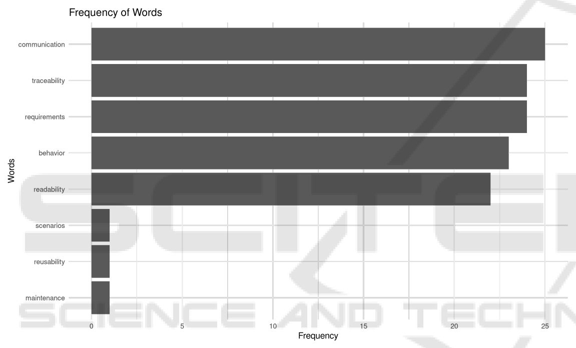

Figure 4: Benefits to adopting BDD.

The aspects highlighted by Dan North remain concerning the use of BDD, these being communication and readability (North, 2006). It is also interesting to note that the other three aspects mentioned appeared with a high number of mentions, so it is possible to infer that BDD also has as positive characteristics the assertiveness in the behavior expected by the system to meet the elicited requirement in addition to the traceability of such requirements.

Furthermore, it is worth highlighting the response of one of the participants regarding the benefits of using BDD, namely: "Reusing the use of similar scenarios in test automation." With this, the effectiveness of using BDD is to help professionals identify the similarity between automated tests, where it is possible to infer the saving of time for the professional and money for the company through this statement.

Q5 - What Are the Main Difficulties Encountered for the Adoption of BDD? According to Figure 5, it is possible to see that the main difficulties mentioned were lack of experience and writing. Lack of knowledge can be linked to the other factors mentioned as negative points, so if there is no user experience, it becomes challenging to maintain a positive result in different aspects.

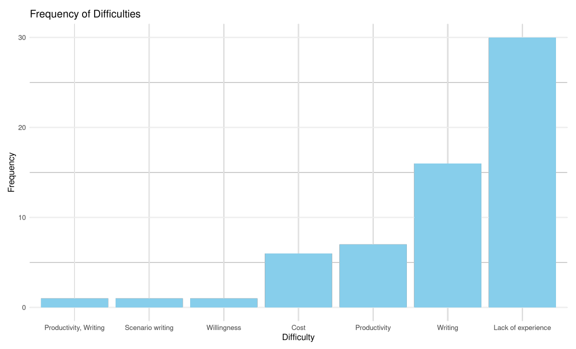

Figure 5: Difficulties in adopting BDD.

It is worth highlighting the response of one of the participants regarding the difficulties in adopting BDD, as follows:

Understanding how it is done, as BDD must be used as a behavioral reference for development using the three friends for writing to occur. Thinking about BDD for any test is not BDD; it is already a test scenario using Gherkin. Many people confuse Gherkin with BDD.

Therefore, it is possible to infer the need for an initial meeting of the "3 friends" where the requirements can be elicited appropriately, aiming for the behavior expected by the software. The meeting aims to illustrate the necessary behaviors of software so that their respective acceptance scenarios are outlined later, according to the participant's response.

Another participant stated the difficulty: "*Learning curve to understand and apply the concept efficiently and effectively.*" Therefore, it is essential to have the necessary understanding inherent to adopting BDD since, if misapplied, the benefits arising from this framework cannot be used.

Finally, the last answer to be highlighted is:

I believe that a "pre-concept" was generated because it was used little or in a certain way incorrectly. When we look at implementation with BDD, there is only an additional layer where more verbose words are written for more understanding, and it does not make execution take longer or cause configuration hangs. I've seen people work, for example, with Cypress and Cuca, which is highly costly *because Cypress itself ends up blocking the reuse of the cucumber.*

This answer may be linked to both cost and lack of experience. It is possible to infer that when choosing other tools to complement the use of BDD, if chosen incorrectly, the budget and quality of software delivery can be compromised, causing losses instead of benefits.

According to Fig. 6, one can see the graphs referring to questions 6 to 12.

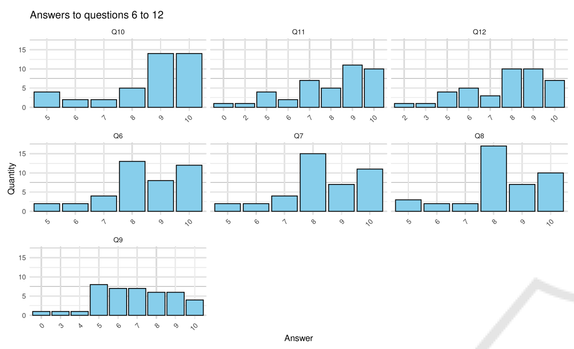

Figure 6: Answers to questions 6 to 12.

These data will be discussed in the following Subsections.

Q6 - On a Scale of 0 to 10, How Much Is (*Gherkin*) Understandable? According to the graph referring to Q6, most respondents chose 8, 10, and 9, respectively, and the rest chose 5 or more. In this way, it sheds light that the BDD language is considered understandable, and it can be inferred that the use of *gherkin* remains positive in the face of recurring work activities.

Q7 - On a Scale of 0 to 10, Considering the Purpose of the BDD Regarding Readability, How Much Does BDD Achieve in this Regard? According to the graph referring to Q7, all respondents chose the value 5 or more, with the highest number of votes being 8, 10, and 9, respectively. Thus, it is inferred that the initial objective of BDD outlined by Dan North is successfully maintained.

Q8 - On a Scale of 0 to 10, Considering the Purpose of the BDD Regarding Communication, How Much Does BDD Achieve in this Regard? According to the graph referring to Q8, all respondents chose the value 5 or more, with the highest number of votes being 8, 10, and 9, respectively. The communication aspect also maintains quality concerning the use of BDD. This factor may be related to the fact that there is a meeting of the "3 friends" so that the requirements and the time for delivering the releases are planned.

Q9 - On a Scale of 0 to 10, How Much Does BDD Perform Faster Delivery? According to the graph referring to Q9, it can be seen that there was a distribution between the scores, ranging from 0 to 10 among the respondents. This factor can be linked to the professional's experience, so the more experience they have, the faster delivery can be carried out.

Q10 - On a Scale of 0 to 10, How Much Does BDD Perform to Deliver with Higher Quality? According to the graph referring to Q10, it is clear that most respondents opted for values 9 and 10, assuming delivery effectiveness with the use of BDD. Additionally, all other participants responded with 5 or more.

Q11 - On a Scale of 0 to 10, How Practical Is BDD to Perform Updates in the Code ? According to the graph referring to Q11, the highest concentration of respondents was between 9 and 10. This aspect can be linked to the fact that the professionals who carry out the code also carry out the documentation in the correct way so that when the code needs to be updated due to a problem, for example, with documentation appropriately done, it will make the work easier, saving resources such as time and money.

Q12 - On a Scale of 0 to 10, How Much Does BDD Contribute to the Relationship to the Company's Economy when Having Living Documentation? According to the graph referring to Q12, responses ranged from values 2 to 10, obtaining their highest concentration between values 8 and 9 with an equal number of respondents and value 10 shortly after that. The relationship between the savings that the company can generate through dynamic documentation is due to the need to document the code correctly, as there is always support in case of a software problem.

5 DISCUSSION

According to Fig. 7, when analyzing questions 6 to 12, it can be identified that all of them have an average above 7.5 in the answers. Only Q9 and Q11 have discrepant values with answers below grade 5.

According to Fig. 8, the relationship between the participants' experience and understanding of BDD can be observed, with a positive correlation of 0.238661. In the Experience line, there are five

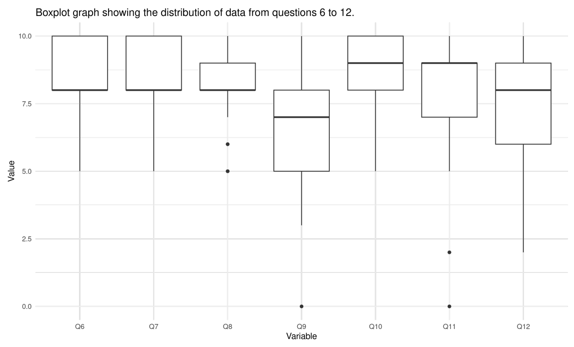

Figure 7: Distribution of answers to questions 6 to 12.

columns with values from 1 to 5, where 1 is less than 2 years; 2 is equivalent to between 2 and 5 years; 3 is between 5 and 7 years; 4 is between 7 and 10 years; and, finally, 5 is equivalent to more than 10 years.

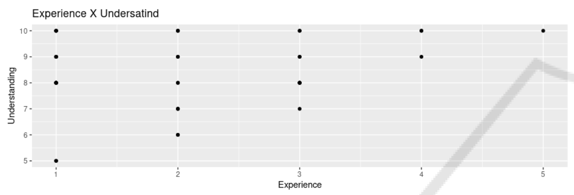

Figure 8: Correlation between experience and understanding.

Analyzing the correlation between experience time and perceived difficulty in using a framework is essential to understanding the impact of developers' experience on the effectiveness of using computational tools. This study calculated the correlation coefficient between the developers' experience time and the perceived difficulty in using the framework as 0.238661. This value suggests a weak positive correlation between these two variables, indicating that, in general, an increase in experience time is associated with a slight increase in perceived difficulty in using the framework.

Although this correlation is statistically significant, its magnitude is relatively low, suggesting that factors other than length of experience may have a more substantial influence on developers' perception of difficulty in using the framework. Therefore, additional studies are necessary to promote a better understanding of the relationship between developers' experience and the effectiveness of using the framework, considering a more comprehensive range of variables and development contexts.

Identifying this correlation between experience and understanding can contribute to the study carried out by Silva and Fitzgerald (Silva and Fitzgerald, 2021), regarding the understanding of BDD so that in this new study, it can be observed that experience directly impacts the understanding of using BDD, in a way that can lead to future problems such as poorly elicited requirements, for example, which could have been mitigated initially if there had been the necessary understanding inherent to the use of the framework.

Regarding experience and readability, a positive, but very weak, correlation of 0.09434189 was obtained, as shown in Figure 9. This correlation between time spent using the BDD framework and the readability objective, as its creator, Dan North, outlined, suggests a positive but weak association between these two variables.

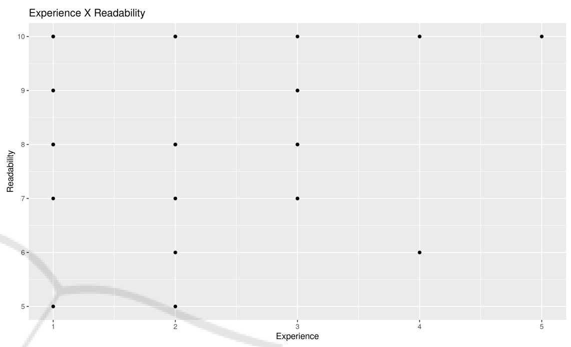

Figure 9: Correlation between experience and readability.

This implies that, in general, an increase in BDD usage time may be associated with an improvement in the readability of tests and software specifications, as recommended by BDD. However, it is essential to note that the correlation is relatively low, indicating that factors other than usage time may significantly impact readability. Therefore, to maximize the effectiveness of BDD in promoting code and specification readability, it is critical to consider not only usage time but also other aspects of the software development process and BDD adoption.

Fig. 10 shows the relationship between experience and communication, with a weak positive correlation of 0.2447518.

Experience also directly impacts the communication factor, so the weak positive correlation demonstrates that based on the knowledge obtained by professionals, communication can be improved to assist in delivering the final product, also acting as an essential factor in the process.

Making a correlation between data collected, it is clear that the level of experience and speed of delivery, as shown in Fig. 11, presents a weak positive correlation of 0.1963602.

One can observe that the most significant relationship between both aspects was between columns 1 and 3, representing the most important number of professionals who, even with little experience using

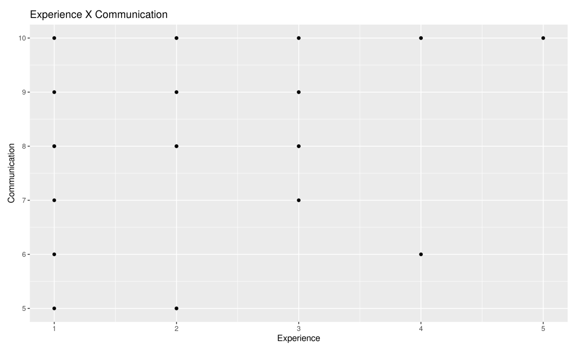

Figure 10: Correlation between experience and communication.

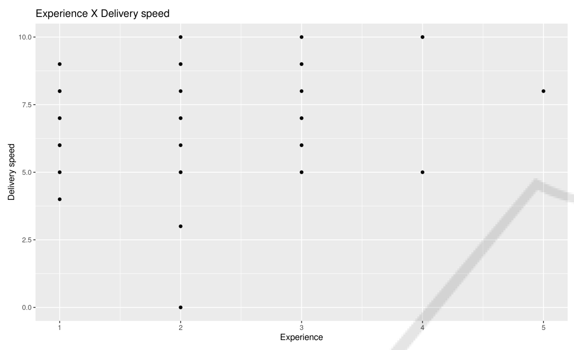

Figure 11: Correlation between time of experience and speed of delivery.

BDD, can identify that the framework has a fast delivery.

Regarding the relationship between experience time and delivery quality, despite the relationship being positive, it is a weak correlation with a value of 0.06831631, as can be seen in Fig. 12.

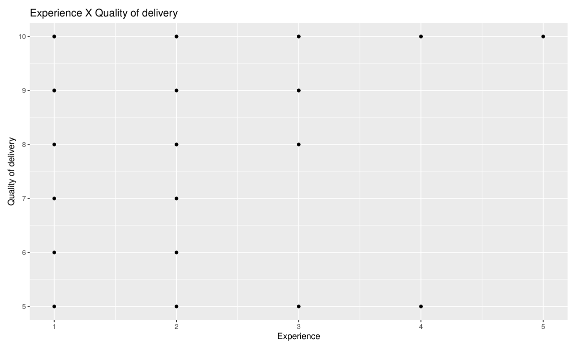

Figure 12: Correlation between experience time and delivery quality.

The correlation of 0.06831631 between time using the BDD framework and quality software delivery suggests a positive but weak association between these variables. Although the correlation indicates that an increase in BDD uptime may be loosely related to an improvement in the quality of software delivery, it is essential to note that this association is relatively low. This implies that factors other than BDD usage time may significantly influence the quality of software delivery, such as the team's expertise, the quality of requirements captured in tests, and the effectiveness of adopted agile development practices.

In the Experience line, columns 1 to 5 represent the experience time, as explained previously. There is a good distribution among respondents, ranging from the professional with the least experience and who sees a low-quality delivery to the professional with the most experience and sees a better quality of delivery. Furthermore, there is a concentration of respondents with low experience who also identify quality in delivery, as shown in the first three columns of the graph.

Therefore, to achieve superior software delivery, it is crucial to consider BDD uptime and adopt a holistic approach encompassing multiple aspects of the software development process and BDD adoption. Next, the length of experience was analyzed about document updating, as shown in Fig. 13. There is an almost insistent correlation, despite being positive, of 0.03743474.

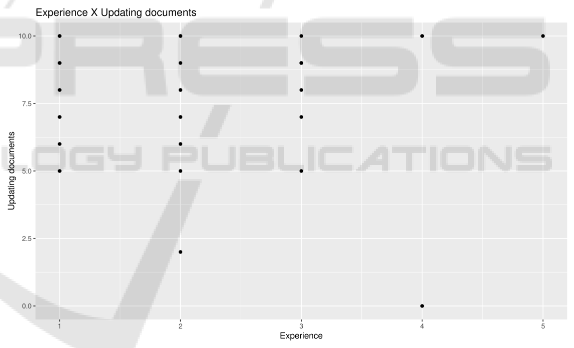

Figure 13: Correlation between length of experience and document updating.

There is a concentration of respondents in the first three columns so that even with little experience, professionals can understand the effectiveness of BDD concerning document updating. Finally, regarding the relationship between experience and savings, there is a negative correlation, although weak, of -0.06944453, being the only negative correlation found, as shown in Fig. 14.

This result may be linked to the aspect that initially, the implementation of BDD is seen as a relatively high cost, so if there is the adoption of something new in a company, there is a need for time to adapt, and, therefore, contribute effectively to the economy where the new project is being implemented.

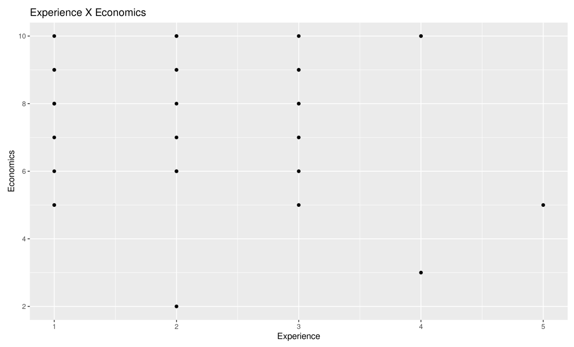

Figure 14: Correlation between experience and savings.

It was possible to characterize and measure the characteristics along with the adoption of BDD to contribute to the work of Rauf and AlGhafees (Rauf and AlGhafees, 2015) regarding the understanding of agile practices. Furthermore, it was possible to advance the knowledge regarding the state of practice in which BDD finds itself.

Although all correlations were low, all were positive except for the correlation between experience and economy. In this way, it is clear at which points BDD can prove to be stronger and at which points there is a need for greater attention, demonstrating the need for knowledge about its scope.

6 CONCLUSION

With the data obtained in this study, it is possible to observe the point of view of 43 professionals who use BDD in their work activities to characterize this framework, making it possible to advance the understanding regarding BDD and its strengths and weaknesses inherent to its adoption.

It was inferred that it was necessary to have experience using BDD to achieve the potential expected by the framework, so its adoption has benefits such as quality in delivery and readability. The lack of experience in using BDD directly impacts the performance of work activities, so the more experience is gained in using the framework, the fewer situations will be considered harmful regarding its adoption.

Therefore, some suggestions for future work could be carrying out a case study with teams that do not yet use BDD in their work activities to validate the information obtained in this study regarding the adoption of BDD; also experiment with two teams, one that has less experience in using BDD and the other more experienced, to correlate the professionals' point of view regarding this framework.

ACKNOWLEDGEMENTS

Shexmo Santos would like to thank CAPES for the financial support (Bolsa de mestrado process no. 88887.888613/2023-00).

REFERENCES

Bruschi, S., Xiao, L., Kavatkar, M., et al. Behavior-Driven Development (BDD): a Case Study in Healthtech. In *Pacific NW Software Quality Conference*, pages 1–12.

Kitchenham, B. and Pfleeger, S. L. (2002). Principles of survey research: Part 5: Populations and samples. *ACM SIGSOFT Software Engineering Notes*, 27(5):17–20.

Mashiko, Y. and Basili, V. R. (1997). Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement. *J. of Systems and Software*, 36(1):17–32.

Moe, M. M. (2019). Comparative Study of Test-Driven Development (TDD), Behavior-Driven Development (BDD) and Acceptance Test–Driven Development (ATDD). *International Journal of Trend in Scientific Research and Development*, 3:231–234.

North, D. (2006). Introducing BDD. https://dannorth.net/introducing-bdd/.

Pereira, L., Sharp, H., de Souza, C., Oliveira, G., Marczak, S., and Bastos, R. (2018). Behavior-Driven Development Benefits and Challenges: Reports From an Industrial Study. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pages 1–4.

Rauf, A. and AlGhafees, M. (2015). Gap Analysis Between State of Practice and State of Art Practices in Agile Software Development. In *2015 agile conference*, pages 102–106. IEEE.

Ribeiro dos Santos, S., Rodriguez, G., and Gomes Rocha, F. (2024). Adopting behavior-driven development (bdd) in software development: a multivocal review. *Memorias de las JAIIO*, 10(2):14–27.

Santos, S., Pimentel, T., Rocha, F. G., and Soares, M. S. (2024). Using Behavior-Driven Development (BDD) for Non-Functional Requirements. *Software*, 3(3):271–283.

Shaw, M. (2003). Writing Good Software Engineering Research Papers. In *25th International Conference on Software Engineering, 2003. Proceedings.*, pages 726–736.

Silva, T. R. and Fitzgerald, B. (2021). Empirical Findings on BDD Story Parsing to Support Consistency Assurance Between Requirements and Artifacts. In *Evaluation and Assessment in Software Engineering*, pages 266–271.

Soares, M. S. and Vrancken, J. L. M. (2011). A Framework for Multi-layered Requirements Documentation and Analysis. In *Proceedings of the 35th Annual IEEE International Computer Software and Applications Conference, COMPSAC 2011, Munich, Germany, 18-22 July 2011*, pages 308–313. IEEE Computer Society.