

Behavior-Driven Development applied in Agile Teams of a Financial Institution

Emanuel Galhardo de Oliveira¹, George Marsicano²

¹Computer Science Department – University of Brasilia (UnB)
Brasilia – DF – Brazil

²UnB Gama Faculty – University of Brasilia (UnB)
Brasilia – DF – Brazil

emanuel.oliveira@aluno.unb.br, georgemarsicano@unb.br

Abstract. *A premise for the efficiency of software developers is clarity on understanding the requirements, as well as the business people maintain the same mutual alignment. Behavior-driven Development (BDD) proposes based on a specific scenario, perceives a problem and correctly directs a solution. This case study shows how BDD helps three agile teams from a Financial Institution to deliver a complex credit platform solution, which at first had critical flaws to bring the product, and how they overcame it. In scenarios where a more focused understanding of a problem is required, it was observed that BDD brought a positive influence for the performance of these teams. While maintaining mature ways of working by improving communication, a mutual understanding of the product added a direction regarding priorities, bringing fast flow with quality. Given the good results, this institution intends to extend the use of this technique to other teams that support critical issues.*

1. Introduction

In this paper, behavior-driven dynamics are investigated in agile projects and the effects of rapid feedback on team performance. During 2021, the performance of three agile teams belonging to a financial institution was analyzed, whose challenge is the development of a complex white label credit platform solution. After a few sprints, critical performance problems were identified about the quantity and quality of the deliveries. The executive board called for an emergency solution. Technical leaders among these teams proposed the use of BDD, and an enabling team was called in to guide and improve the applicability. The goal was to primarily help business personnel in macro definitions on how the product should behave. In the sequence there were specific studies to consolidate a first version of the roadmap of these teams.

2. Background

The concept of WHAT IS BDD was created in 2003 by developer Dan North [North 2006], as a response to another programming methodology, *Test-driven development* (TDD)¹. On the experience of instructing IT professionals in learning TDD, a

¹Test-driven development, or TDD, is a software development technique in which the test cases are defined previously to specify and prove what the code can do.

methodology based on exhaustive systems testing, he noted that his students raised pertinent questions, such as where to start testing or what didn't need tests. How to abstract the real scenario of a functionality. In this perception Dan North began to formulate the concept of BDD, which combines agile development strategies and encourages collaboration between professionals from different areas in software creation [North 2006]. Thus, it is possible to contemplate a more cohesive view of a product and develop solutions according to the expectations for the final user, meeting the business goals.

3. Related Work

In this section, some works related to the theme of this experience report are presented, aiming to present to the reader some objectives and results that have been identified in the literature, regarding the use of BDDs.

Amna et al. [Amna and Poels 2022] and Sarangee et al. [Sarangee et al. 2022] emphasize how BDD facilitates the work of the agile team by bringing clarity in the business definition for the development of features based on user stories. Kortum et al. [Kortum et al. 2019] explored and reported human factors-driven performance on software projects. It shows how exploratory analysis can improve interpretations of the relationships between, for instance, social conflict, communication, and gathering behavior. Alferez et al. [Alferez et al. 2019] describes the impact of interactions during meetings on positive mood afterwards, using empirical evaluation in an industrial case study in the financial domain, which was driven by early but strong evidence of the feasibility and benefits of behavioral specifications.

Scandaroli et al. [Scandaroli et al. 2019] conclude its benefits collected from both adoptions, such as increased self and mutual confidence for both the remote and local teams, organic documentation of product features provided by the BDD scenarios, decreased communication gap between remote and local teams and improved product quality and stability. There is empirical reliability demonstrating that in other case studies using the BDD technique in agile teams got the same positive results.

4. Methodology

Regarding the context of the unit of analysis, data were collected on those three agile teams. Data types were chosen based on the hypothesis that there is a positive correlation between the use of BDD and team productivity. During the observation, it was also identified that the culture of tests in teams that consolidate their backlog without the use of BDD are reactive in the performance of covering tests in the code, different from the previous performance of tests applying TDD, of those that work with BDD. To evidence and compare the observations in agile teams, two views with and without BDD are proposed. Given these views the data was captured in the following items per sprint:

- *Volume of deliveries by story points;*
- *Technical debts generated by story points;*
- *Number of meetings with Product Owner (PO)²;*
- *Average lead time by user story;*
- *Number of unattended user stories;*
- *Level of developers satisfaction.*

²Product Owner, or PO, is a strategic role responsible for representing the interests of the customer on the development team.

5. Results

In order to create a white label credit platform solution, functionalities must be modularized to allow reuse. These lines of credit will be created on the platform with combinations desired by the business area, which will strategically act based on the needs that market demand. With a possibility of dynamically creating these credit products, this Financial Institution will have great advantage. In addition to the internal use of these functionalities, based on being a white label solution, it is possible to share certain features to partner firms expanding customer's targets.

Concern about the three agile teams are integrated by six developer members each. They are experienced on scrum based software development methodology plus engineering practices. Due to the weight of the complexity of the subject, well-defined user stories are critical to avoid rework and more often technical debts. This three teams took on the new credit solution in the second half of 2020 using agile methodology and eXtreme programming (XP) techniques. The monitoring of the amount of deliveries occurred during the year 2021. In that year, the use of the BDD technique was started in February.

On all teams the velocity histogram shows a clear progress in the mobilization to work with the BDD. There was an increase in *volume of deliveries by story points* of around 60%, considering the entire annual period. With more constant alignments before the planning, *technical debts generated by story points* were reduced due the maturity of functionalities definition. Also perceptible was the *number of meetings with PO* that decreased substantially, with rare occurrences of once in a month for alignments, which was mostly to treat in the dailys.

In test management surveys it currently costs 25% of *average lead time by user story* to create relative test cases. Specific on this three teams were identified about 12% lower of time spent on developing test cases. Based on scenarios already written and more cleared definition of the proposed functionalities by the PO. The *number of unattended user stories* in the sprint was another factor that practically nullified itself. From the application of the BDD, the situation of bringing more stories beyond those that were previously agreed in the planning was recurrent. Collecting a total of 18 responses of the three teams, where the *level of developers satisfaction* was asked, if the BDD facilitates their activities, the results were 88% of great influence on the work.

6. Discussion

Through the researched articles, other large corporations were identified that use the BDD technique to bring better targeting of their products. This leads us to understand that the results presented confirms what the literature brings.

M. Irshad et al. [Irshad et al. 2021] based on research questions in 2021 relate five main benefits related to the use of BDD in large-scale software projects. These benefits are; understanding of a business aspect of requirements, improved quality of requirements, a guide to system-level use-cases, reuse of artifacts in large-scale projects, and help for test organization. It also ensures that the time saved at the end of the development cycle can be added to the exploratory testing, thus increasing the product's quality.

Inside a large scale software organization that is developing business support systems for telecommunication, M. Irshad et al. [Irshad et al. 2022] evaluation in 2022

showed encouraging results and the practitioners' feedback was positive. They revealed that higher diversity in BDD specifications results in higher coverage of product features during the validation activities. Another research provided by Güncan et al. [Güncan and Onay Durdu 2021] evolving 73 participants from several organizations discovers that by using the BDD approach, business units are included in the creation of scenarios, thus facilitating communication among software developers, analysts, and customers and providing more information on system behavior than just technical details. In addition, communication among all stakeholders is carried out in a healthy manner. The results identified are very similar compared to the case study in this article.

7. Conclusion and Future Work

BDD is shown to facilitate software development for agile teams, allowing members to better understand the features under development, improving alignment with the right priorities in the backlog. Based on this case study in highly complex problems, BDD had a great influence on overcoming challenges in product delivery. Given the strategic directions within the financial institution, an expansion of new teams is expected to apply new insights to confirm whether the technique will have the same results.

References

- Alferez, M., Pastore, F., Sabetzadeh, M., Briand, L., and Riccardi, J.-R. (2019). Bridging the gap between requirements modeling and behavior-driven development. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pages 239–249. IEEE.
- Amna, A. R. and Poels, G. (2022). Ambiguity in user stories: A systematic literature review. *Information and Software Technology*, 145:106824.
- Güncan, D. and Onay Durdu, P. (2021). A user-centered behavioral software development model. *Journal of Software: Evolution and Process*, 33(2):e2274.
- Irshad, M., Börstler, J., and Petersen, K. (2022). Supporting refactoring of bdd specifications—an empirical study. *Information and Software Technology*, 141:106717.
- Irshad, M., Britto, R., and Petersen, K. (2021). Adapting behavior driven development (bdd) for large-scale software systems. *Journal of Systems and Software*, 177:110944.
- Kortum, F., Klünder, J., and Schneider, K. (2019). Behavior-driven dynamics in agile development: The effect of fast feedback on teams. In *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pages 34–43. IEEE.
- North, D. (2006). Introducing bdd, better software magazine.
- Sarangee, K., Schmidt, J. B., Srinath, P. B., and Wallace, A. (2022). Agile transformation in dynamic, high-technology markets: Drivers, inhibitors, and execution. *Industrial Marketing Management*, 102:24–34.
- Scandaroli, A., Leite, R., Kiosia, A. H., and Coelho, S. A. (2019). Behavior-driven development as an approach to improve software quality and communication across remote business stakeholders, developers and qa: two case studies. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, pages 105–110. IEEE.