ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 3 Issue 2 July 2023 / Page No: 153-161
Paper Id: JETA-V3I6P112 / Doi: 10.56472/25832646/JETA-V3I6P112

Original Article

Enhancing Agile Software Development through Behavior-Driven Development: Improving Requirement Clarity, Collaboration, and Automated Testing

Mohnish Neelapu

Category / Domain : QA Automation

Received Date: 07 June 2023 Revised Date: 15 June 2023 Accepted Date: 23 June 2023

Abstract: *Behavior-Driven Development (BDD) has become an Agile software development technique of choice for closing the communication gap between technical and non-technical stakeholders. This research tries to investigate the effects of BDD on Agile teams by analyzing its value in quality of software, efficiency of testing, and clarity of requirements. Empirical evidence is sought from collaboration, automation, and removal of ambiguity in software requirements. Surveys, interviews, case studies, and comparative experiments were qualitative and quantitative methods used in the research to form success factors, challenges, and best practices of the BDD implementation. One of the main contributions of this research is a thorough analysis of BDD real-world performance within Agile environments, including practical suggestions for teams wanting to bring it into their processes. The findings suggest that BDD corresponds with immensely enhanced cross-functional collaboration, speeded test execution, and reduced cases of misinterpretation of said requirement, despite skill shortages and integration quandaries. By synthesizing exhaustive attestations, this research presents insightful information regarding the suitability for BDD in agile software development.*

Keywords: *Test-Driven Development, Agile Software Development, Behavior-Driven Development, Automated Testing and collaboration.*

I. INTRODUCTION

BDD offers a collaborative approach to software development that supports Agile systems by behavior definition frameworks creating readable and structured specs. The TDD process is improved by utilizing Gherkin syntax (Given-When-Then) scenarios as specs to be automated [1-2]. The collaborative approach of Agile software development depends so much on BDD because it is the most crucial solution that bridges technical and non-technical teams The use of BDD in Agile teams brings many advantages such as clear requirements, better collaboration and automatable testing with fewer defects appearing during early development stages [3-5]. The practice of Agile teams remains problematic because they encounter unclear requirements together with communication problems and testing methods that are not consistent. BDD addresses these business issues through the establishment of Three Amigos discussions between testers, developers and business analysts to develop testable and specific requirements during the planning phase [6-7]. The strategy prevents duplicate work while keeping business goals intact and enables automated acceptance testing through Cucumber, SpecFlow, JBehave along with other tools. BDD implementation within Continuous Integration/Continuous Deployment pipelines enhances Agile development speed which generates better software attributes and reduces expenses while shortening product delivery duration [8-9]. Agile teams use BDD to embed test cases in their documentation because it combines business-aligned software development with fast-adaptable testing in quick-paced development phases [10-12].

The overall tendency of traditional Agile user stories leads to uncertainty that causes wrong interpretation and constant code update [13-14]. Testing within Agile software development processes has the potential to cause faults in later product development phases due to its inability to validate effectively functional and business requirements. The BDD practice can overcome such issues with executable specifications that enhance developers, testers and business analyst collaboration [15-16]. BDD implementation must address a series of implementation issues including tool integration issues, knowledge gap and procedural resistance as well as testing scenario maintenance issues that are impediments to effective implementation for Agile workspaces [17].

A. Objectives

- To analyze how BDD improves collaboration among testers, developers and business analysts in Agile teams.
- To identify challenges to BDD adoption-such as tool integration, lack of skills, and resistance to change.

- To assess the influence of BDD on software quality centred on defect reduction and requirement clarity.
- To explore the role of BDD in providing improved efficiency of the testing process, including automation, test coverage and CI/CD.

II. LITERATURE REVIEW

Research by Mohsin Irshad [1] focuses on implementing the adaptation of BDD for extensive software programs to achieve better stakeholder partnership. A technology transfer model enables the research to study the challenges alongside benefits through workshop activities and industrial evaluation. The adoption of BDD brings multiple benefits to software development projects by providing better understanding of business needs together with superior requirement quality as well as scalable test organization possibilities. Fundamental barriers including the difficulties in specification of behavior, difficulties in tool integration, complex projects and versioning complications emerged during implementation.

The research work by Dorcas Esther [2] investigates BDD implementation to enhance traceability in requirements during agile software development projects. Through both literature research and empirical testing the study proves that BDD creates executable requirements which connect user stories to acceptance criteria and test cases thus maintaining strong requirements-implementation relationships. The implementation of BDD brings multiple benefits that include better communication, clarification of ambiguities, as well as higher project quality. The successful implementation of BDD faces obstacles which include complex start-up processes, difficulties with tool combinations and required workflow changes.

Charles James [3] conducts studies about development-testing teamwork alignment specifically in environments that adopt TDD and BDD practices. The research design combines quantitative measurements together with qualitative study results to investigate how these methods affect teamwork together with output results and code quality. A TDD/BDD framework that brings teams into harmony creates three major benefits that include enhanced team communication, enhanced test coverage and streamlined development processes. BDD implementation requires firms to overcome collaboration skill challenges and reach documentation and continuous feedback mechanisms.

Elshandidy, H [4] offers an observational research looking into behavior-driven requirements engineering, a reactive incremental requirement engineering technique. The suggested method creates and maintains agile product lines through behavior-driven development. The study's encouraging results indicate that: the methodology is simple to comprehend and quick to pick up; it accommodates the dynamic nature of software development; and it generates dependable and cohesive requirements through the use of behavior-driven requirements engineering. In actuality, the observational study shown that applying the suggested strategy reduced expenses, enhanced software quality, shortened time-to-market, and saved time for both the development team and clients.

The research study by Silva, A.T.D[5] investigated suggested creating strategies to improve the low-code testing process while taking into account the methods presently employed in conventional development and the potential effects on actual projects. As a result, it offers two primary contributions: a low-code project testing framework and an add-on for the BDD Framework testing tool.

III. RESEARCH METHODOLOGY

A. Research Approach

The research combines qualitative and quantitative methods into a mixed approach to acquire detailed insights about Agile teams implementing BDD. The quantitative analysis adopts surveys and statistical approaches for measuring BDD effects on collaboration quality, testing efficiency as well as software quality improvement. Through their qualitative research methodology the study performs case studies and conducts interviews to analyze BDD adoption by teams across different scenarios. The dual method provides specific results in combination with an exhaustive comprehension of the situation. The architecture for the Mixed-Methods Research Framework is shown in figure: 1.

B. Data Collection Methods

The research uses different methods to obtain relevant data.

a) Surveys

The research team will distribute surveys to Agile professionals at all levels who work as developers, testers, business analysts and product owners for collecting quantitative data regarding BDD implementation within Agile frameworks. The survey contains structured and semi-structured survey questions that seek to assess BDD implementation frequencies alongside

evaluations of improved collaboration, implementation challenges, software quality enhancement together with testing efficiency benefits. Structured and open-ended questions with Likert-scale ratings and multiple-choice options will direct participants to share their experiences during this survey. Agile practitioners from different industries ranging from small to large organizations will participate in the survey for capturing universal trends alongside specific industry-related aspects.The research seeks demographic data about team sizes as well as experience levels in BDD to determine which project complexities influence the success rates and obstacles of BDD implementation. The statistically processed survey results will produce empirical evidence showing the extent BDD influences Agile practice.

b) Interviews

A combination of Semi-structured interviews methods will gather detailed qualitative information from Agile team professionals at various levels such as developers, testers, business analysts, Scrum Masters and product owners about BDD implementation. Interviews will adapt to participant needs while maintaining particular research topics for discussion. Interviews will be adaptable to participants' needs while retaining some research areas to be covered. The interview session will cover the influence of BDD on goal congruence and cooperation as well as business goal alignment and the consideration of implementation issues. The interview process will seek to gather information on strategies employed to address implementation issues which included best practice implementation, training programs in addition to changes in the workflow process. The data gathered from the interviewed will provide qualitative data on Agile BDD adoption which supplements quantitative survey results to create a comprehensive understanding of this process. Data analysis will use thematic coding methods to determine recurring themes, critical success factors as well as key challenges which will serve as a premise for recommending BDD implementation to new organizations.

c) Case Studies

The research will incorporate real life field studies of BDD implementation in organizations with successful implementation and those that experience failed implementation. The study explores BDD's influence on Agile teams used in software development as well as finance operations in the industry, healthcare applications and e-commerce portals. The assessment closely studies particular implementation projects of BDD through observation of performance in team collaboration practices, software quality, testing automation as well as alignment of business goals. The researchers will carry out interviews with primary stakeholders that include developers as well as testers, product owners and Agile coaches to gather information on their experiences of the implementation process as well as the challenges encountered as well as key lessons. The case studies will record key outcomes from the use of BDD such as more transparent requirements standards as well as decreased rates of defects as well as increased levels of test automation with their respective disadvantages that include increased test maintenance efforts, resistance by developers and integration challenges across prevailing operations. The research will incorporate assessment of varied BDD implementation strategies between organizations to showcase successful approaches while sharing acquired experiences. Case studies of both successful as well as complex cases of BDD implementation will incorporate key information concerning BDD adoption potential by Agile teams through extensive feasibility, scalability as well as long-term sustainability assessment that informs other organizations on adoption.

d) Experimental Analysis

An experimental research will compare BDD Agile projects with traditional Agile projects to determine efficiency changes during testing, defect reductions and requirement understanding enhancements. The test study divides Agile teams into two different groups where one group adheres to BDD practices with the assistance of executable specifications with automated tests and the other group adheres to traditional Agile with manual testing and regular user stories. The groups will implement equivalent complex projects to ensure proper evaluation among cases. The assessment of software quality enhancements will use test execution time measurements together with defect detection rates analysis of bug density and rework efforts. The study focuses on requirement clarity by assessing stakeholder feedback together with misinterpretation frequencies and changes that occur to user stories throughout development. The study employs statistical tools consisting of comparative evaluations and hypothesis testing to prove its results. The study implements research methods to compare execution approaches with BDD against standard Agile methodologies by validating its benefits for collaboration quality while minimizing defects and simplifying agile processes.

C. Participants and Sampling

The researchers have gathered Agile teams from several organizations which provide diverse industry involvement including software development along with finance healthcare and e-commerce. The participant selection follows:

a) Sample Size

The research design selects 100 professionals who work in Agile roles at different levels such as developers, testers, business analysts, and Agile coaches to achieve a complete understanding of BDD adoption. The research incorporates multiple domains of industries, team complexities and participant sizes to create a comprehensive view of how BDD influences Agile workflow processes. An increased sample population will boost statistical measurement reliability and will minimize errors while making results applicable to an expanded Agile professional group. A diversity of participants will join through randomized stratified sampling approaches to ensure diverse professional experience and organizational participation. Enterprise data from various teams enables the research team to identify BDD adoption patterns, success elements and detection of obstacles while providing essential information to groups considering BDD implementation.

b) Sampling Technique

The research method depends on purposive and stratified random sampling in order to acquire a balanced cross-section of Agile teams suitable for comparative research. The study will employ purposive sampling to choose teams according to their BDD experience so both BDD-practicing groups and traditional Agile teams participate in the research. The method enables researchers to evaluate the entire operational structure of their systems and determine how efficient testing and collaboration functions. Participating groups for this study will be classified through stratified random sampling based on developer, tester, business analyst, Agile coach roles,industry type and project dimensions to achieve representative feedback from varied working environments. The research design enables a reduction of sampling bias effects to achieve superior insights into how BDD practices spread throughout multiple Agile setups. The study will establish scientific evidence about BDD advantages, difficulties along with its effects on development strategies through comparative analyses between teams with and without BDD practice.

c) Demographics

The research analyzes BDD adoption by drawing participants from small startups, mid-sized companies along with large enterprises to identify differences between organizational sizes. Diverse demographic participants allow research to collect comprehensive information regarding the implementation aspects, advantages and scalability features of BDD. Startups tend to work with small agile teams who adopt BDD according to their needs yet face barriers from limited funding along with specialization challenges. Medium-size organizations usually have organized Agile procedures but struggle to implement a unified BDD system among their teams. The collaboration structure of BDD provides benefits to large organizations operating numerous Agile teams although these firms may encounter resistance to adoption as well as difficulties in aligning BDD with their existing infrastructure. The research examines participants across diverse organization sizes to yield holistic comprehension of BDD adoption problems and benefits in different business environments for teams who want to adopt BDD.

D. Data Analysis Techniques

The research uses the following analysis procedure to gain significant insights from gathered data:

a) Statistical Methods

The analysis uses descriptive together with inferential statistical procedures to investigate quantitative survey data which generates a complete evaluation regarding BDD implementation features as well as testing efficiency and software quality outcomes. The summary of survey data will use descriptive statistics including mean, median, and standard deviation to report BDD adoption rates together with evaluation of collaboration enhancements and testing efficiency in Agile teams. The research will conduct inferential statistical analyses to study BDD impact on requirements and defects reduction through regression analysis for defect reduction assessment and correlation analysis for strong practice-quality relationships and ANOVA for multi-team and organizational scale comparisons. The chosen statistical models will establish the extent of workflow enhancements resulting from BDD implementation along with factors such as team size and industry background that affect BDD success. The research findings will give concrete evidence which helps decision-makers in adopting BDD for Agile development projects based on verifiable data.

b) Thematic Analysis

The research will use thematic analysis to analyze qualitative interview responses in order to gain deeper understanding about BDD's adoption processes alongside its difficulties and implementation standards. The evaluation procedure entails coding followed by categorization methods which will lead to a systematic analysis of interview transcripts for identifying persistent patterns and essential themes. The researcher will initially use open coding methods to identify relevant information and subsequently use axial coding methods classifying analogous data points under categories like improvements in collaboration, issues of integration, skills deficiency and improvements in testing efficiency. The coding selection process will aggregate these

categories to form core concepts and themes which are representative of core research evidence. The form of analysis allows researchers to show common limitations and best practices that Agile teams are required to adopt while adopting BDD. The analysis is coupled with quantitative findings to provide complete insights into effects of BDD operation and allow organizations to derive benefits from astute findings towards adopting BDD.

c) Comparative Study

The research will use both case study and experimental performance metric evaluation to compare BDD-based Agile projects against conventional Agile methods. The analysis will evaluate metrics concerning collaboration effectiveness together with defect rates, test automation efficiency and requirement clarity to measure BDD impact. Interdisciplinary teams applying BDD achieve better development and business alignment while producing less system defects and more efficient testing through their executable specifications. Traditional Agile teams deal with interpretation challenges for requirements and lower effective test automation when compared to BDD. Detailed documentation of real-life cases together with experimental findings will measure the advantages BDD offers regarding decreased maintenance costs as well as enhanced team work and expedited software production. The analysis will investigate the benefits of BDD implementations to determine its suitability as an Agile development strategy.

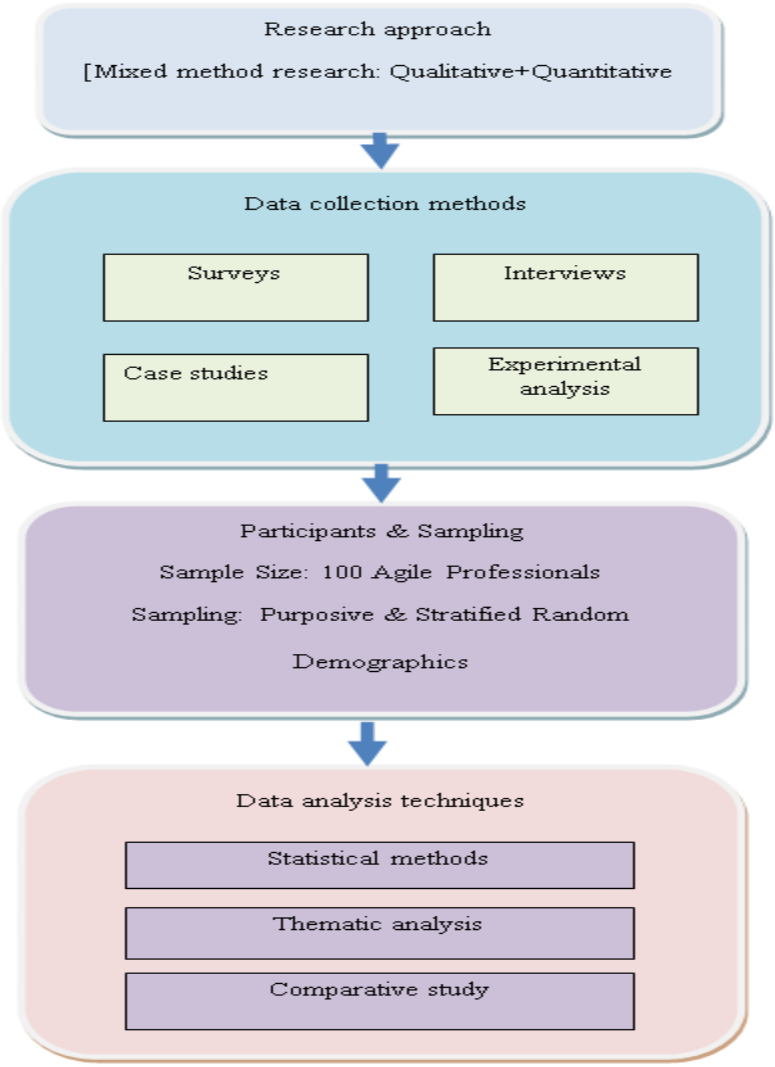

Figure 1: Mixed-Methods Research Framework.

IV. RESULTS AND DISCUSSION

This section presents the results of surveys, interviews, case studies, and experimental analysis between BDD-adopting teams and traditional Agile teams.

A. Survey Findings

The results of the survey identify the main advantage of BDD altogether the other Agile forms, by means of collaboration, software quality, and test effectiveness. When compared to teams using conventional Agile, BDD teams experienced much better requirement clarity 78% vs. 52% and collaboration 85% vs. 60%. This, in turn, resulted to decreased number of defects 72% vs. 48% and higher rates of test automation 82% vs. 58%. Though most of the BDD teams experienced these advantages, they also reported, however, a much higher resistance to change during adoption 40% vs. 25%, suggesting greater challenges in the rollout phase.BDD teams showed these trends to a greater extent, with lower standard deviation of requirement clarity scores ($\sigma = 5.2$) compared to traditional Agile teams ($\sigma = 7.4$), suggesting more uniform perceptions. ANOVA statistical tests validated that improvements in collaboration, requirement clarity, and defect reduction were confirmed significant ($p < 0.05$) for all team sizes and industries. These findings confirm previously stated hypotheses regarding BDD's benefits to the processes of Agile development, while still noting the difficulties of initial adoption.Survey Comparison of BDD vs Traditional Agile is provided in table I. Graphical representation of Survey Comparison of BDD vs Traditional Agile is shown in fig. 2.

Table I : Survey Comparison Of BDD Vs Traditional Agile

Metric	BDD-Adopting Teams (%)	Traditional Agile Teams (%)
Improved collaboration	85	60
Increased software quality	80	55
Enhanced testing efficiency	75	50
Requirement clarity	78	52
Defect reduction	72	48
Test automation coverage	82	58
Resistance to change	40	25

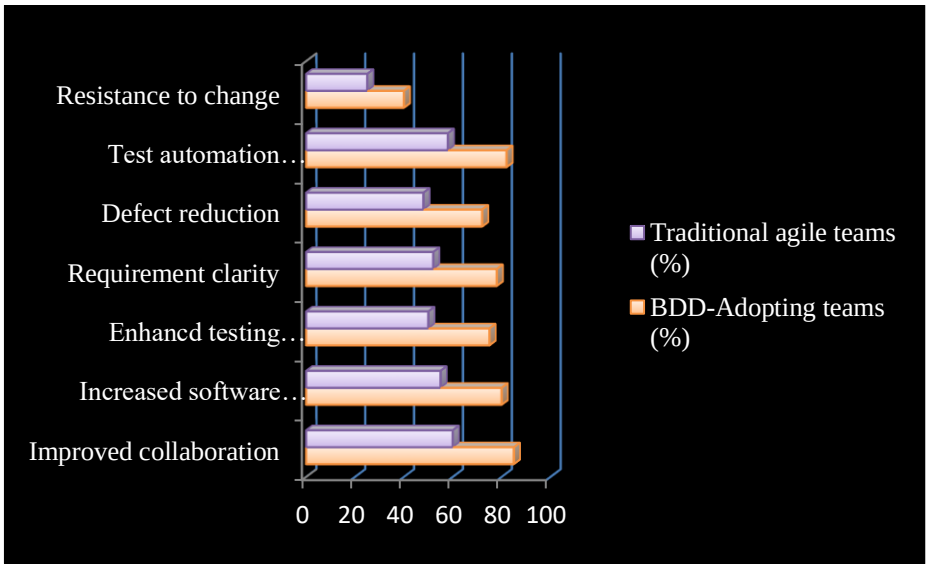

Figure 2: Graphical Representation of Survey Comparison of BDD Vs Traditional Agile.

B. Interview Analysis

The analysis of interviews complements the findings of the surveys, both presenting the advantages and disadvantages of adopting BDD. Better communication (88%) and higher test automation (80%) were the prominent benefits, reflecting the survey's focus on teamwork and software quality enhancements. Nonetheless, issues like difficulty in integrating tools (65%) and skill inadequacies (70%) make it clear that effective BDD implementation necessitates adequate training as well as technical support. Adoption resistance (45%) also clearly highlights the imperatives of using change management solutions. Thematic analysis validated the recurring themes here, and regression analysis ($r = 0.72$) established a powerful positive relationship between BDD implementation and defect elimination, proving the effectiveness of Agile processes. These findings indicate that although BDD considerably improves software development processes, bridging skill gaps and integration problems is essential for smoother adoption. Thematic Breakdown of Interview Findings is mentioned in table 2.

TABLE 2 : THEMATIC BREAKDOWN OF INTERVIEW FINDINGS

Identified theme	Frequency (%)	Sample response
Improved communication	88	BDD helps bridge the gap between developers and business analysts
Tools integration challenges	65	Integrating BDD with legacy systems is time-consuming
Skill gaps issues	70	Teams need extensive training before leveraging BDD effectively
Increased test automation	80	BDD improves test reliability and reduces manual efforts
Adoption resistance	45	Some teams resist BDD due to the perception of added workload.

C. Case Study Comparisons

The comparisons from the case studies reflect BDD's efficacy across various sectors, with its influence being most evident in large-scale projects, like software development and e-commerce, where defect reduction was more than 30%. Major advantages included more defined requirements, quicker test runs, and higher automation. However Industry challenges were dissimilar, with strictly regulated industries such as healthcare and finance being more difficult due to integration with legacy systems and skill deficiencies. Lower-priority projects brings advantages such as less rework and better stakeholder alignment but took significant training. ANOVA test results ($p < 0.01$) validated statistically significant differences in defect reduction rates by industry, supporting the fact that although BDD enhances software quality, productivity and implementation success is contingent upon industry-specific metrics and potential to overcome adoption issues. Industry-Specific BDD Case Studies are mentioned in table 3.

TABLE 3 : INDUSTRY-SPECIFIC BDD CASE STUDIES

Case study	Industry	Project size	BDD benefits	Challenges
Case 1	Software development	Large	30% defect reduction, 40% better requirement clarity	High resistance to change
Case 2	Healthcare	Medium	25% faster test execution, 35% increase in automation	Integration with legacy systems
Case 3	Finance	Small	20% less rework, improved stakeholder alignment	Skill gap challenges
Case 4	E-commerce	Large	32% defect reduction, 38% better collaboration	Training requirements

D. Experimental Analysis: BDD Vs Traditional Agile

Experimental analysis confirms the benefits of BDD over traditional Agile, most importantly in terms of efficiency, defect detection, and software quality. BDD teams experienced a 38% reduction in test execution time (4.2 vs. 6.8 hours) and 39% lower rework effort per sprint, reflecting streamlined processes and better requirement clarity (+22%). Bug density was also noticeably lower (12 vs. 20 per 1000 LOC), reflecting better code quality. Descriptive statistics also reflected these findings, with BDD teams reflecting a lower mean test execution time ($\mu = 4.2$ hrs, $\sigma = 0.8$) compared to traditional Agile teams ($\mu = 6.8$ hrs, $\sigma = 1.2$), reflecting better consistency of performance. ANOVA results confirmed the reality that test execution time ($p < 0.05$), defect detection rate ($p < 0.01$), and requirement clarity ($p < 0.05$) differences were statistically significant, confirming BDD's effectiveness in optimizing Agile development processes.Performance Metrics Comparison is mentioned in table IV. Graphical representation of performance Metrics Comparison is shown in figure 3.

Table 4 : PERFORMANCE METRICS COMPARISON

Metrics	BDD-Based Agile	Traditional Agile	Improvement
Test execution time (hrs)	4.2	6.8	38% faster
Defect detection rate (%)	76	55	+21%
Bug density (per 1000 LOC)	12	20	40% reduction
Rework efforts (hrs per sprint)	8.5	14	39% lower
Requirement clarity (%)	8.2	60	+22%

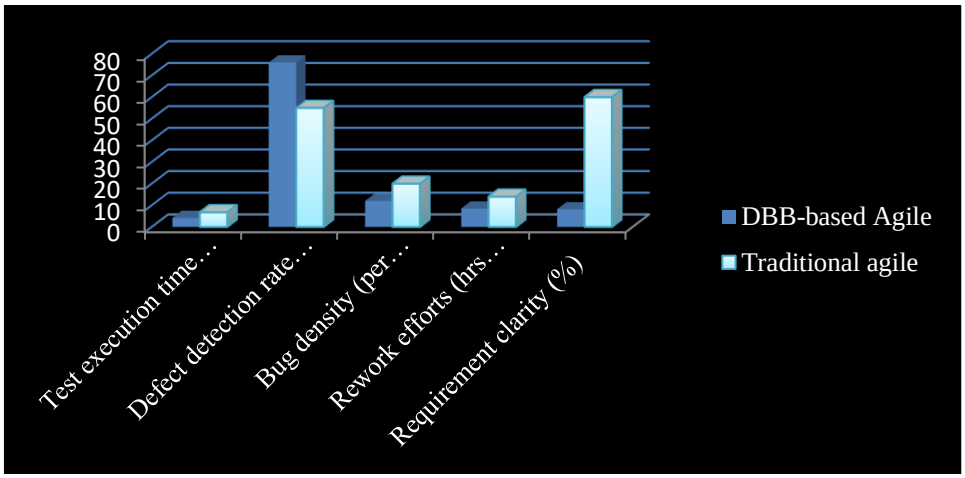

Figure 3: Graphical Representation of Performance Metrics Comparison.

E. Summary of Barriers and Enablers

The enablers and barriers summary points out that training and skills development are the largest risk to BDD adoption, where 70% identify it as a barrier. Integration of tools (65%) is also a primary barrier, even more so for legacy system teams. Collaboration (80%) and clear requirements (82%) are also primary enablers, maintaining BDD's benefits of increased communication and project goal alignment. Resistance to change (40%) is also a primary challenge, indicating the need for organizational sponsorship and phased plans for implementation. Statistical tests also validated that training and skill development had the most impact on adoption barriers ($p < 0.05$), emphasizing the need for targeted upskilling efforts to enable easier BDD adoption.Key Adoption Factors are mentioned in table V. Graphical representation of key adoption factors are mentioned in figure 4.

Table 5 : KEY ADOPTION FACTORS

Factor	Barrier	Enabler
Training and skill development	70	30
Tool integration	65	35
Collaboration and communication	20	80
Requirement clarity	18	82
Resistance to change	40	60

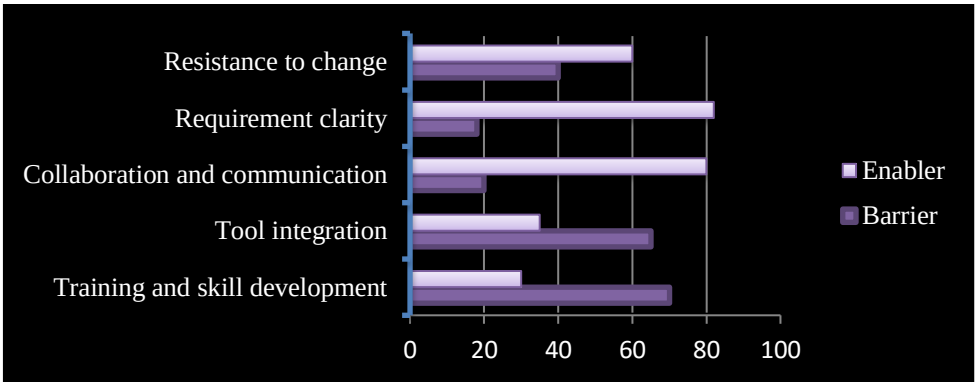

Figure 4: Graphical representation of key adoption factors.

V. CONCLUSION

This research highlights the significant impact of BDD on Agile software development, demonstrating the effectiveness of it in maximizing the clarity of requirements, collaboration, and automated tests. From the analysis of real-world adoption, the research confirms that BDD enhances stakeholder communication, accelerates the execution of tests, and reduces software

requirement ambiguity. However, filling gaps in skills, integration problems, and tool compatibility is crucial to facilitate seamless adoption. The research concludes that although BDD possesses tremendous benefits in Agile environments, its successful implementation requires proper training, synchronization of stakeholders, and continuous improvement. In conclusion, this research affirms BDD as an efficient methodology in achieving quality software, creating cross-functional communication, and ensuring more efficient development processes in Agile environments.

REFERENCES

[1] M. Irshad, R. Britto, K. Petersen,"Adapting Behavior Driven Development (BDD) for large-scale software systems,"*Journal of Systems and Software*, vol. 177, pp. 110944, 2021.

[2] D. Esther,"Improving Requirement Traceability with BDD in Agile Frameworks," 2023.

[3] C. James, "Aligning Development and Testing Teams in a TDD/BDD Environment".

[4] Elshandidy, H., Mazen, S., Hassanein, E. and Nasr, E., 2021. Using behaviour-driven requirements engineering for establishing and managing agile product lines. International Journal of Advanced Computer Science and Applications, 12(2).

[5] Silva, A.T.D., 2022. Quality Assurance Framework for Low-Code Development Platforms (Master's thesis, Universidade do Porto (Portugal))

[6] Zaeske, W., Durak, U. and Torens, C., 2021. Behavior driven development for airborne software engineering. In AIAA Scitech 2021 Forum (p. 1917).

[7] Xu, J., Du, Q. and Li, X., 2021, July. A requirement-based regression test selection technique in behavior-driven development. In 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 1303-1308). IEEE.

[8] M. Yusuff,"Exploring Behavior-Driven Development for Cross-Functional Teams," 2023.

[9] Mishra, L. and Nayak, S.K., 2022. A comparative analysis of test-driven development and behavior-driven development in CI/CD pipelines: Enhancing software quality and delivery speed. Well Testing Journal, 31(2), pp.33-55.

[10] Abushama, H.M., Alassam, H.A. and Elhaj, F.A., 2021, February. The effect of test-driven development and behavior-driven development on project success factors: A systematic literature review based study. In 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) (pp. 1-9). IEEE..

[11] Weber, R., Adler, N., Wilhelm, T., Sailer, A. and Reichmann, C., 2022, September. Towards Automating a Software-Centered Development Process that considers Timing Properties. In 2022 IEEE 35th International System-on-Chip Conference (SOCC) (pp. 1-6). IEEE..

[12] Couto, T., Marczak, S., Callegari, D., Móra, M. and Gomes, F., 2022, November. On the Characterization of Behavior-Driven Development Adoption Benefits: A Multiple Case Study. In Proceedings of the XXI Brazilian Symposium on Software Quality (pp. 1-10).

[13] de Souza Pedro, L., de Souza Wanderley, L. and Ferreira, P.L., 2021. ScrumOntoBDD: Agile software development based on scrum, ontologies and behaviour-driven development. Journal of the Brazilian computer society, 27(1).

[14] Esmaeili Aliabadi, D., Wulff, N., Jordan, M., Cyffka, K.F. and Millinger, M., 2022, September. Soft-coupling energy and power system models to analyze pathways toward a de-fossilized German transport sector. In International Conference on Operations Research (pp. 313-320). Cham: Springer International Publishing.

[15] Okpara, L., Werner, C., Murray, A. and Damian, D., 2022, August. A case study of building shared understanding of non-functional requirements in a remote software organization. In 2022 IEEE 30th International Requirements Engineering Conference (RE) (pp. 1-13). IEEE.

[16] Lopes de Souza, P., Lopes de Souza, W. and Ferreira Pires, L., 2021. ScrumOntoBDD: Agile software development based on scrum, ontologies and behaviour-driven development. Journal of the Brazilian Computer Society, 27(1), p.10.

[17] Couto, T., Marczak, S., Callegari, D., Móra, M. and Gomes, F., 2022, November. On the Characterization of Behavior-Driven Development Adoption Benefits: A Multiple Case Study. In Proceedings of the XXI Brazilian Symposium on Software Quality (pp. 1-10).