iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44745

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Behavior Driven Development using Gauge Framework

M. Shamanth[1], Prof. Prashanth K[2]
[1]*Student, Department of MCA, RV College of Engineering®, Bengaluru, India*
[2]*Assistant Professor, Department of MCA, RV College of Engineering®, Bengaluru, India*

Abstract: ***Behavior Driven Development (BDD) focuses on the behavior of the system or the business point of view and captures the business specific requirements. Many BDD frameworks implement a strict syntax like Gherkin, this study proposes a solution to remove this by using Gauge Framework. The BDD Gauge Framework will be analyzed and studied in-depth. The drawback of this study is that a big industry level testing isn't feasible hence will need to be dependent on external research papers. The emerging opportunity here is that Gauge Framework can be considered as a BDD tool with its unique inbuilt capability of parallelization or scaling of tests, data driven testing, no strict restricted syntax and also enhances agile development. Key findings of this study will be how Gauge Framework improves testing with better defect detection while maintaining good method coverage for various business scenarios and also how it can be used as a BDD framework by reducing manual effort. This paper will also determine the performance with respect to other competitive frameworks in the market like Cucumber, RSpec, SpecFlow.***

Keywords- Gauge framework, BDD, Software testing, Test Automation, Behavior Driven Development

I. INTRODUCTION

Behavior Driven Development (BDD) is a software/application development idea which enhances the agile software development process [1]. The goal here is to identify user behavior and capture business specific requirements and convert them into test specifications. This process in general brings collaboration between involved parties like developers, product owners and management teams. It allows the members to use a shared pool of tools and processes. The main tool behind this development is the use of a Domain Specific Language (DSL) [2]. In this type of development normal English or simple language is used instead of complex code to express the behavior. BDD is a refinement of Test Driven Development (TDD). It is a way of formalizing TDD to make it more focused on delivering enterprise software. In both Test Driven Development and Behavior Driven Development, the test cases will be written before any code is developed. BDD makes sure that Enterprise software is expected to last for years as it presents a user perspective. Its process is translating behavior language into a specialized language like Gherkin and then a parsing library such as Cucumber invokes unit testing framework. The figure 1 below shows an example test case for a login scenario using Gherkin language which uses a strict syntax.

Feature: File Upload

Scenario 1: Testing valid file upload to AWS S3

Given a valid file with data to upload
When user uploads the given valid file to destination
Then the file gets uploaded to S3 bucket

Fig. 1: Example of Gherkin language test case 1

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

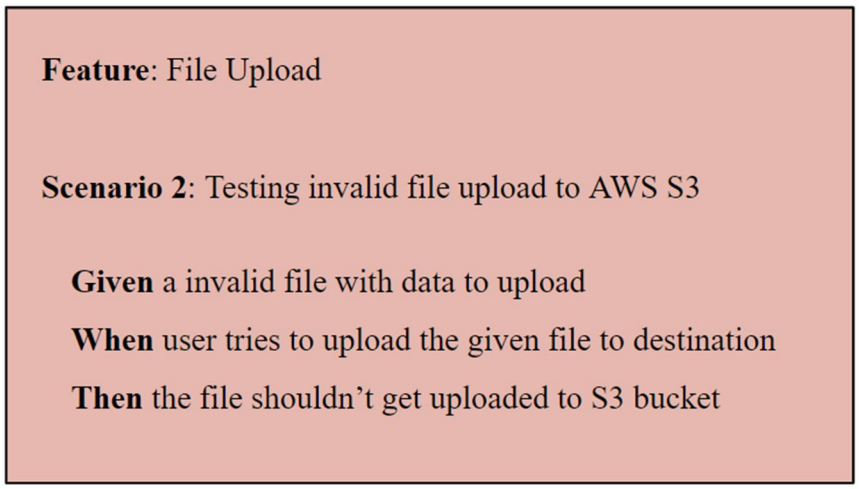


Fig. 2: Example of Gherkin language test case 2

The frameworks which use Gherkin language will need to define the scenarios using the specified keywords like Given, When and Then imposing strict syntax. This is where Gauge Framework is proposed to be used which uses natural language for writing down test specifications. The business readability factor increases significantly when Gauge Framework is used as compared to other frameworks. The scenarios can be written and that itself can act as a living document.

II. LITERATURE SURVEY

Behavior Driven Development has achieved higher customer satisfaction as compared to Test Driven Development. This also consumes more time for development and has been observed to have lower customer acceptance than BDD. This paper investigated the study results from previously published research papers and articles which were published in the years between 1999 till 2020 [3].

Description in a paper suggested that there was an increased positive impact when BDD was used, increasing collaboration among team members, and less negative impacts, like difficulties in writing unit tests. This paper had investigated the BDD on agile teams' software development where they had used this process in real life. The outcome was that more positive results than negative effects [4].

Best practices and examples of developing and writing automated test scripts for testing in simple natural language requirements using the Gauge framework is explained. Paper discusses the scenario where the BDD framework enforces a pretty strict syntax and analyzes the same with the easily readable syntax of Gauge Framework [5].

Comparison and the deep study of TDD is done with respect to Behavior-Driven Development and with Acceptance TDD(ATDD) while studying the 3 types of development process and its working. It concludes that BDD evolved from TDD eliminating its shortcomings enhancing the overall software development process [6].

Benefits of communication are investigated with respect to BDD. Also, the impact of alignment on stakeholders, the design in software, along with changes in code or refactoring of code, and fixing of bug are analyzed. Two experiences applying the BDD are explored and explained [7].

Characteristics of Behavior Driven Development are analyzed with relevant literature and available tools along with their features in the paper Behavior Driven Development: Tools and Challenges authored [8] by R. K. Lenka, S. Kumar and S. Mamgain,

Test-Driven-Development has greater issue/defect identification ability whereas lower methods coverage than Test Last Development. The study conducted in an industrial context resulted in test driven development being more effective than TLD. This study is beneficial to make future research using behavior driven development as it should be even more effective than test driven development [9].

There are various challenges which are faced during the development phase of test automation which need to be overcome to prevent issues. Auditing the key challenges faced by agile practitioners and different strategies to overcome those challenges while practicing test automation are investigated [10].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Details about the benefits of test driven development is majorly because of the fact that Test Driven Development like scenarios encourage and improve very focused perfected, steady finite steps which are responsible for improving the flow. The paper discusses the benefits of testing first or testing last and provides an insight on whether it will have an impact with these two different approaches [11].

An approach of developing test automation software with closed collaboration with the product development team was discussed in the paper Developing software product and test automation software using Agile methodology authored by A. K. Sultanía [12].

III. DISCUSSION

Behavior Driven mindset is used in software development planning where the team will clarify the requirements helping with better understanding of the objectives. According to research conducted [13] during the beginning stages of a software development, the usage of BDD has seemed to have a positive impact on the project specification and requirements. Also, the development team and the project managers have given positive feedback and they have felt an improved feature understanding. The study seems to have found that BDD could benefit the members involved in the development of the software. Since the features are understood in a better way the quality of implementation will also be improved and it will in turn result in better documentation. Usage of BDD has encouraged development team and product managers to collaborate in a better way, which is proven by a thesis written by Moraes [14].

In Gauge Framework, automated test cases are written in a simple natural language like English. These sentences are used as executable statements and the related test code is written in the development language, java in this case. The delivered output could be different from what was expected if there is no proper communication between the development team and the business representatives. This can be improved if the scenarios are properly defined during the initial stage. The product quality can be significantly improved and so will the stability of the product.

IV. ANALYSIS OF GAUGE FRAMEWORK

Gauge Framework gives provision of tasks like logging, reporting, and having property files thereby reducing the setup time. Figure 3 shows a testing example using Gauge framework where the scenario of a file being uploaded to s3 is shown. Here the behavior can be defined in very simpler terms than the given-when-then format.

```
# File Upload Feature

table: resources/Db/Resource_File.csv

## Testing valid file upload to AWS S3

* Upload a file <destFilePath> <sourceFilePath> <fileName> to S3
```

Fig. 3: Example file upload scenario using Gauge Framework

```
@Step("Upload a file <destFilePath> <sourceFilePath> <fileName> to S3")
public void uploadFile(String destFilePath, String sourceFilePath, String fileName) throws InterruptedException {
        DisplayUtil.displayBanner( str: "Running upload file");
        String destination = destFilePath + fileName;
        String source = sourceFilePath + fileName;
        S3FileOperations.copyFile(source, destination);
}
```

Fig. 4: File upload scenario implementation in java using Gauge Framework

Figure 4 shows the implementation of the above scenario using java code. This could also be written in other languages like Javascript, C#, Python or Ruby. This same method could be reused for a different step as well, this capability of code reuse is provided inbuilt in Gauge Framework.

©IJRASET: All Rights are Reserved

TABLE 1
COMPARISON BETWEEN BEHAVIOR DRIVEN DEVELOPMENT FRAMEWORKS

Features/BDD Frameworks	Gauge	Cucumber	SpecFlow	Rspec
Syntax	Markdown	Gherkin	Gherkin	Ruby
Passing large data files	Yes	Possible, with custom parsing logic	Yes	Possible, with custom parsing logic
Data Driven support	Yes	Yes, using scenario outline	Yes, using scenario outline	Yes, using custom logic
Parallel execution support	Default support	Yes, using plugins	Yes	Doesn't support
Language support	Multi-languages	Multi-languages	.NET	Ruby
Environment specification support	Pass environment variables without writing related code	Need to write custom logic	Need to write custom logic	Need to write custom logic
Reporting	Inbuilt support and very easy to read	Uses a reporting service	Inbuilt support for reporting	Uses a reporting service

The advantage which gauge poses here is that it has many features inbuilt most of which can be made available in other frameworks using external plugins. Even though it has plenty of features it is lightweight and works well as intended. Since Gauge Framework doesn't impose a syntax while writing the specifications or scenarios they can be written as per the companies set of formats and make the development better.

V. CONCLUSION

In this paper, the impact of BDD in agile software development teams has been analyzed using previous research conducted by other papers. Testing a software completely is not possible and it takes more than 50% of the project budget hence a better modularized automation testing will be required to be in place to keep everything in check.

Creating scenarios using strict syntax like gherkin could be a challenge, hence with the usage of Gauge framework the BDD scenarios can be written seamlessly without much of the readymade strict given-when-then format and still consume the advantages of behavior driven development.

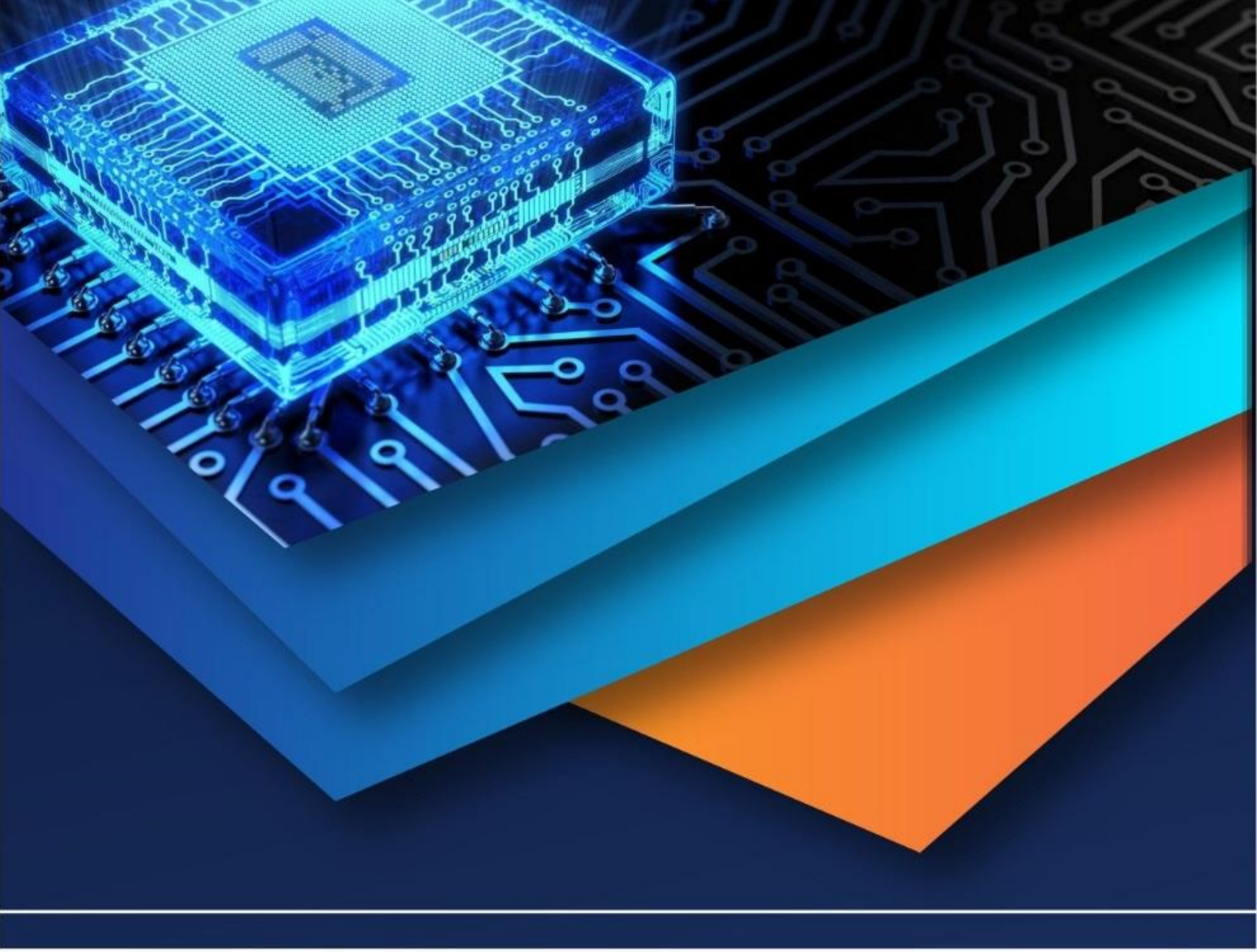
REFERENCES

[1] Marko Anastasov, "Behavior-Driven Development", Internet: https://semaphoreci.com/community/tutorials/behavior-driven- development, May 26, 2022
[2] Bas Dijkstra, "Making Test Frameworks Readable: The Domain Specific Language", Internet: https://medium.com/@bas.dijkstra54/making-test-frameworks-readable-the-domain-specific-language-c154c9a9abcb, September 08, 2020
[3] Abushama, Hisham M., Hanaa Altigani Alassam, and Fatin A. Elhaj. "The effect of test-driven development and behavior-driven development on project success factors: A systematic literature review based study." 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). IEEE, 2020.
[4] Nascimento, Nicolas, et al. "Behavior-driven development: A case study on its impacts on agile development teams." Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. 2020.
[5] Garousi, Vahid, et al. "Test automation with the Gauge framework: Experience and best practices." International Conference on Computational Science and Its Applications. Springer, Cham, 2020.
[6] Moe, Myint Myint. "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 (2019): 231-234.
[7] Scandaroli, André, et al. "Behavior-driven development as an approach to improve software quality and communication across remote business stakeholders, developers and QA: two case studies." 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE). IEEE, 2019.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[8] Lenka, Rakesh Kumar, Srikant Kumar, and Sunakshi Mamgain. "Behavior driven development: Tools and challenges." 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018.
[9] Tosun, Ayse, et al. "On the effectiveness of unit tests in test-driven development." Proceedings of the 2018 International Conference on Software and System Process. 2018.
[10] Tyagi, Sulabh, Ritu Sibal, and Bharti Suri. "Adopting Test Automation on Agile Development Projects: A Grounded Theory Study of Indian Software Organizations." International Conference on Agile Software Development. Springer, Cham, 2017.
[11] Fucci, Davide, et al. "A dissection of the test-driven development process: Does it really matter to test-first or to test-last?." IEEE Transactions on Software Engineering 43.7 (2016): 597-614.
[12] Sultania, Ashish Kumar. "Developing software product and test automation software using Agile methodology." Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT). IEEE, 2015.
[13] Nascimento, Nicolas & Santos, Alan & Sales, et al, "Behavior-Driven Development: A case study on its impacts on agile development teams", Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, 2020.
[14] Lauriane Moraes. 2016. An Empirical Study on the Use of BDD and its Support to Requirements Engineering. Master's thesis. Pontifical Catholic University of Rio Grande do Sul, Brazil

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)