

# Research on Unifying Business and Testing with Cucumber Automation

Sayli Subhash Chorge

Student, Department of Masters of Computer Applications

Late Bhausahab Hiray S. S. Trust's Hiray Institute of Computer Application, Mumbai, India

**Abstract:** *The synchronization of business goals with software testing procedures is essential for producing high-quality, dependable software solutions in today's quickly changing business environment. A major difficulty today is bridging the gap between testing teams and business stakeholders. This study examines how Cucumber automation can be used to integrate business and testing practices. A well-liked behaviour-driven development (BDD) framework called Cucumber offers a distinctive method that makes it possible to collaborate effectively, improve communication, and guarantee the delivery of software solutions that are in line with company needs. This article discusses the advantages, difficulties, and effective implementation techniques of Cucumber automation in bridging the gap between businesses and testing through an analysis of actual case studies and industry best practices. The results highlight the significance of having a collaborative mind-set, having clear communication, and having a well-defined framework for using Cucumber automation. The research comes to a close with suggestions for future work and potential improvements in leveraging Cucumber automation to further strengthen the fusion of business and testing practices, ultimately resulting in higher software quality and client satisfaction*

**Keywords:** Cucumber, BDD, Gherkin, Cucumber Automation, unifying business and testing, Eclipse

## I. INTRODUCTION

Collaboration between business stakeholders and testing teams is essential in the constantly changing world of software development if high-quality software solutions are to be delivered. However, the historical gap between these two groups frequently obstructs good communication, resulting in errors, extra effort, and delays.

The understanding, sharing, and validation of software requirements are revolutionised by Cucumber, a widely used behaviour-driven development (BDD) framework. Cucumber makes it easy for business stakeholders and testing teams to collaborate by offering a single language and framework, dissolving organisational silos and encouraging a common knowledge of software needs. Business stakeholders can use the Gherkin language to convey their expectations in a systematic manner thanks to the power of Cucumber automation. These human-readable, executable specifications act as a link between the testing procedure and the business domain. Utilising these specs, testing teams may design automated tests that determine whether the programme satisfies the specified criteria, removing any room for ambiguity and guaranteeing transparency throughout the development process.

In addition, Cucumber plays a significant part in fostering cooperation, providing quick feedback loops, and supporting agile and iterative development approaches. The study paper will also go over Cucumber's ability to integrate with different testing frameworks and tools, such Selenium. Through the combination of Selenium's extensive automation capabilities with Cucumber's expressive features, this integration enables testing teams to take full advantage of the benefits of both frameworks.

## II. LITERATURE REVIEW

A key component of effective software development has long been understood to be the alignment of business objectives and testing procedures. The advantages and effects of Cucumber automation in fostering better

communication between business and testing teams have been examined in a number of studies. In a case study they did in a sizable software development company, Singh and Jain (2018) discovered that the deployment of Cucumber automation enhanced communication, decreased rework, and raised customer satisfaction. The researchers emphasised how Cucumber's structured natural language specifications made it easier for stakeholders and testing teams to communicate and create a shared understanding of needs.

In a similar line, Cucumber automation's use in an agile development context was examined by Gupta et al. in 2019. According to their research, Cucumber improved software development process transparency by allowing stakeholders to articulate requirements in a language they could understand. The collaborative nature of Cucumber automation, which allowed for constant input and made sure that the created software was in line with the intended business objectives, was highlighted by the researchers.

Additionally, the ability of Cucumber automation to integrate with various testing frameworks has drawn attention in the literature. Li and Zhang (2020) investigated how Cucumber may be integrated with Selenium, a well-known web automation tool. They discovered that the combination of these frameworks allowed for efficient end-to-end testing and made it easier to carry out acceptance tests with a business-oriented focus. For thorough and effective testing, the researchers emphasised the value of utilising Cucumber's expressive language with Selenium's strong automation capabilities.

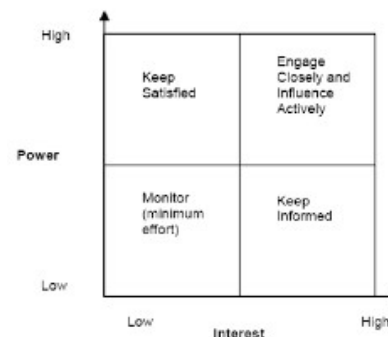
Although Cucumber automation has many advantages, some studies have also discussed the difficulties in using it. Rahman et al.'s (2017) study underlined the value of education and training in ensuring that testing teams and business stakeholders alike thoroughly comprehend the principles and practises of Cucumber automation. To successfully apply and exploit the power of Cucumber in integrating business and testing, the study emphasised the necessity for a collaborative mentality and a change in conventional testing methodologies.

Overall, the body of research is consistent with the idea that Cucumber automation is essential for connecting business and testing. It has been discovered that the framework's natural language specifications, collaboration tools, and integration capabilities promote collaboration, match software development with corporate goals, and boost testing productivity. However, in order to effectively utilise the promise of Cucumber automation, successful

adoption necessitates a focus on training, collaboration, and a change in perspective.

Further research might examine the long-term effects of Cucumber automation on software quality, project success rates, and overall collaboration between business and testing teams as this field of study is still in its infancy. Investigations exploring the integration of Cucumber with cutting-edge technologies like artificial intelligence and machine learning may also shed light on how business and testing may be combined in software development in the future.

### III. PROBLEM DEFINITION



The misalignment between corporate objectives and testing procedures has been a recurring problem in the context of software development. The conventional approaches to gathering and disseminating software requirements sometimes rely on technical documentation and specifications that business stakeholders find difficult to understand. Due to the gap between the testing efforts and the anticipated business outcomes, there are inefficiencies, rework, and decreased stakeholder satisfaction. Additionally, the issue is made worse by the lack of a consistent structure for collaboration and a vocabulary that both business and testing teams can use to communicate effectively.

The power of Cucumber automation is used to solve this issue. A solution is provided by Cucumber, a behaviour-driven development (BDD) framework, which offers an organised and understandable method for expressing software requirements. Cucumber promotes efficient communication, encourages cooperation, and creates a common knowledge of software behaviour throughout the entire development team thanks to its natural language specifications written in Gherkin.

The emphasis is on utilising Cucumber automation as a means of bridging the gap, coordinating software development with corporate goals, and enhancing the general efficacy and efficiency of the testing process.

#### IV. OBJECTIVE/SCOPE

##### 4.1 Objective

The objective of this research is to investigate and demonstrate the effectiveness and potential of Cucumber automation-

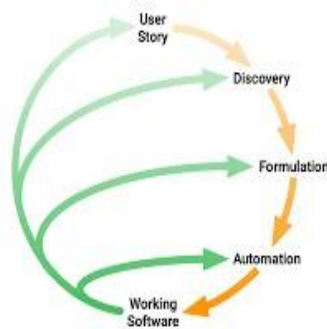
1. Analyze how Cucumber automation affects teamwork and communication between business stakeholders and testing teams: This goal entails examining how Cucumber automation, which offers a common language and structure for expressing software requirements, promotes efficient cooperation and communication. Examining how Cucumber bridges the gap between business stakeholders and testing teams' respective domains and promotes a common understanding of software behavior is the goal of the research.
2. Consider the advantages and difficulties of implementing Cucumber automation in software development projects: Analyzing the benefits Cucumber automation offers to the software testing process, such as improved requirement clarity, decreased rework, and increased stakeholder satisfaction, is required to achieve this goal. The research also attempts to identify and address the difficulties in using Cucumber automation, such as attitude adjustments, training needs, and Cucumber test suite maintenance.
3. Discover effective Cucumber automation implementation techniques and best practices: Examining actual case studies, best practices for the industry, and implementation techniques that have led to effective acceptance and use of Cucumber automation are the main objectives of this purpose. The goal of the study is to pinpoint crucial elements and suggestions for maximizing Cucumber automation's advantages, such as integration with other testing frameworks, efficient test management, and scalability.
4. Examine how Cucumber automation affects software quality and development effectiveness: This objective entails assessing the quantitative and qualitative effects of Cucumber automation on software quality, taking into account elements like defect detection, test coverage, and customer satisfaction. The study also attempts to evaluate how Cucumber automation affects development efficiency, including shortened time-to-market,

quicker feedback loops, and increased productivity among development teams.

##### 4.2 Scope

The scope of the research encompasses several key areas: -

1. Cucumber Automation: In order to bridge the gap between business and testing, the research focuses on examining the capabilities and features of the Cucumber automation framework. Using Cucumber Automation, facilitating communication between business stakeholders and testing teams.
2. Business and Testing Alignment: The study explores the difficulties and possibilities in coordinating testing procedures with corporate goals. It looks at how Cucumber automation may help business stakeholders, such product owners and domain experts, and testing teams collaborate and communicate more easily so that they can make sure the software satisfies the intended business requirements.
3. Collaboration and Communication: The study investigates how collaboration and communication between the business and testing teams are impacted by Cucumber automation. It looks at how Cucumber automation improves communication by giving software behaviour a common language and structure, minimising ambiguity, and promoting productive cooperation throughout the software development lifecycle.
4. Software Quality: The study investigates how Cucumber automation affects the calibre of software. It looks into the role Cucumber automation plays in ensuring that the created software complies with all necessary business criteria and acts as expected. This involves assessing how well Cucumber automation works to find bugs, increase test coverage, and improve software quality overall.
5. Development Efficiency: The study evaluates how Cucumber automation affects the effectiveness of software development. It looks at how implementing Cucumber automation can speed up feedback cycles, reduce rework, and streamline the testing process. Additionally, the research looks at how automation using Cucumber might shorten time-to-market, boost team productivity, and improve overall development effectiveness



Implementation Strategies and Best Practices: The study looks on effective implementation methods and best practises for utilising Cucumber automation. In order to determine the critical elements that contribute to the effective adoption and application of Cucumber automation in integrating business and testing practises, this study looks at real-world case studies, industry experiences, and expert advice.

## V. RESEARCH METHODOLOGY

The research methodology involved following steps to gather data and analyze it-

### 5.1 Research Design

To investigate the subject, the study used a mixed-methodologies strategy that combined qualitative and quantitative methods. This method gave rise to a deeper comprehension of the phenomenon being studied and allowed for a thorough analysis of the research topics.

An online survey that was given to Cucumber automation practitioners, test experts, and business stakeholders was used to gather quantitative data. Closed-ended questions were used in the survey to elicit quantifiable information about respondents' perceptions, experiences, and the effects of Cucumber automation on teamwork, communication, software quality, and development effectiveness.

In-depth interviews with chosen participants were used to acquire qualitative data in addition to quantitative data. Interviews with practitioners and stakeholders who had first-hand experience with Cucumber automation were performed in a semi-structured manner. The interviews covered their implementation of Cucumber automation experiences, difficulties they encountered, and triumphs.

Additionally, a thorough assessment of the literature was done to compile secondary data from credible web sources, books, conference papers, academic journals, and other sources.

### 5.2 Research Questions

The research aimed to address the following key research questions related to our topic

Q. How does Cucumber automation straddle the testing and business worlds?

The goal of this topic was to comprehend the procedures and methods used by Cucumber automation to promote efficient interaction and coordination between business stakeholders and testing teams.

Q. What are the advantages and difficulties of using automation with Cucumber in software development projects?

The advantages of greater productivity, greater test coverage, and better software quality were evaluated in this inquiry, along with the drawbacks of tool setup, learning curve, and organizational reluctance to change.

Q. How does Cucumber automation affect teamwork and communication between testing teams and business stakeholders?

In order to effectively communicate with testing teams and reduce misunderstandings, this study examined how Cucumber automation enables business stakeholders to convey requirements in a structured and accessible style.

Q. What impact does automation using Cucumber have on the effectiveness and quality of software development?

The goal of this topic was to examine how Cucumber automation helps to improve software quality and development efficiency by finding errors early in the development process, guaranteeing adequate test coverage, and optimizing the testing workflow.

### 5.3 Data Collection

The data collection process for the research involved multiple methods to gather both primary and secondary data. These methods included surveys, interviews, and a comprehensive literature review.

#### Surveys:

Those with practical experience using Cucumber automation, testing experts, and business stakeholders were given access to a structured online survey. In order to better understand how Cucumber automation affects working together and communicating with testing teams, the poll included questions. Additionally, it gathered information on the perceived advantages, difficulties, and efficiency of Cucumber automation in integrating business

and testing practices. Quantitative information from the survey responses was available for statistical analysis.

#### **Interviews:**

Selected individuals who have practical experience with Cucumber automation underwent in-depth interviews. The interviews were semi-structured, allowing for a fluid discussion to explore their experiences, difficulties they encountered, and successes they had with regard to putting Cucumber automation into practice. The qualitative data from the interviews captured the subtleties, individual viewpoints, and in-depth understandings of the influence of Cucumber automation on business and testing alignment. For additional analysis, the interviews were audio recorded and transcribed.

#### **Literature study:**

Secondary data were compiled from academic journals, conference papers, books, and reliable online sources as part of an extensive literature study. The review's main objective was to locate pertinent papers, industry standards, and theoretical frameworks pertaining to Cucumber automation, behavior-driven development, software testing, and the coordination of business and testing practices. The results of the literature review assisted the analysis and discussion of the research findings and offered a foundation for comprehending the present state of knowledge in the subject.

#### **5.4 Data Analysis-**

The findings from the quantitative and qualitative analyses were integrated to provide a comprehensive understanding of the research topic. The convergence and divergence between the two types of data were examined to identify overarching patterns and to triangulate the results. This integration of findings ensured a robust and comprehensive analysis of the data collected, providing a rich and multi-faceted perspective on the power of Cucumber automation in unifying business and testing practices.

The analysis process involved rigorous scrutiny of the data, attention to detail, and systematic interpretation of the findings. The results derived from the data analysis formed the basis for the subsequent discussion, conclusions, and recommendations in the research paper. The analytical insights obtained from the research study contributed to advancing knowledge in the field of software development, emphasizing the significance of

Cucumber automation as a valuable tool for bridging the gap between businesses and testing.

#### **5.5 Research Findings**

The results showed that the alignment of business and testing practices was positively and transformative impacted by Cucumber automation. It promoted teamwork, facilitated communication, and enhanced the efficacy and efficiency of the testing procedure. Cucumber automation enhanced a shared understanding, reduced errors, and improved the overall software development lifecycle by uniting business requirements and testing processes.

For businesses considering adopting Cucumber automation, the research findings offer useful information. To fully realize the promise of Cucumber automation in bridging the gap between business and testing, decision-making processes and implementation methods can be informed by the benefits and obstacles discovered.

#### **5.6 Conclusion**

The research's conclusions showed that Cucumber automation has a lot of potential for integrating business and testing practices. Automation using Cucumber can improve software development projects by fostering cooperation, increasing efficiency, and raising software quality. The power of Cucumber automation and bridging the gap between business and testing can be used by organizations with the help of practical recommendations and an understanding of the ramifications, which will ultimately result in more successful software projects and increased stakeholder collaboration.

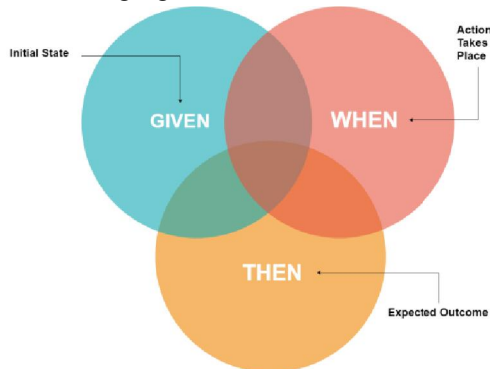
### **VI. ANALYSIS AND FINDINGS**

Cucumber automation helps in unifying business and testing through its distinctive features and technical capabilities. Here's a technical explanation of how Cucumber automation achieves this:

#### **6.1 Behavior-Driven Development (BDD) Approach**

The BDD methodology, which emphasizes cooperation between business stakeholders, developers, and testers, is used by Cucumber. It facilitates the development of Gherkin-synthesized executable specifications in a human-readable manner. Using basic linguistic devices like Given, When, and Then, this syntax enables business stakeholders to structurally communicate their requirements. These specifications act as active

documentation and give the business and testing teams a unified language.



### 6.2 Scenario-based Testing

Cucumber divides tests into scenarios that correspond to particular user interactions or business behaviors. Each scenario is made up of a series of phases with Gherkin syntax definitions. These steps are mapped to the appropriate automation code, which carries out the intended activities and checks that the results are what were anticipated. By coordinating testing activities with business requirements using a scenario-based approach, testers may concentrate on validating essential business functionalities.

### 6.3 Test Automation Framework

As a framework for test automation, Cucumber works flawlessly with many other computer languages, including Java, Ruby, and JavaScript. Testers can create reusable and maintainable automation code by implementing step definitions in their choice programming language. This flexibility enables close collaboration between developers and testers, utilizing each group's specialization in the implementation and upkeep of the automation system.

### 6.4 Business-readable Test Reports

Cucumber produces thorough and easily understandable test reports that give insight into the outcomes of test execution. The steps that succeeded, failed, or are still pending are highlighted in these reports along with the status of each scenario. Stakeholders in the business can examine these reports and learn more about the test coverage and general system behavior. The business and testing teams can effectively communicate, debate test results, and make decisions as a result of this transparency.

### 6.5 Continuous Integration and Delivery (CI/CD) Integration:

Cucumber seamlessly interfaces with CI/CD pipelines, enabling automated Cucumber scenario execution as part of the software delivery cycle. Every time a change is made to the codebase, testers can trigger the Cucumber tests, ensuring that the business requirements are continuously validated. Through early issue detection, shorter feedback loops, and collaboration between development and testing teams, this integration helps the software development lifecycle.

## VII. LIMITATIONS AND FUTURE SCOPE

Although integrating business and testing practices through cucumber automation has many advantages, it also has several drawbacks and potential future research areas. Understanding these constraints and figuring out potential directions for future research might help this discipline advance even further. The following are some restrictions and future potential-

### 7.1 Learning Curve and Skill Requirements:

The initial learning curve associated with Cucumber automation is one restriction. Testers and business stakeholders must learn Gherkin syntax and the underlying automation framework in order to adopt Cucumber. To reduce the learning curve, future research can concentrate on creating more user-friendly technologies, offering thorough training programs, and streamlining the adoption process.

### 7.2 Scalability and Complexity:

In order to handle large and complex projects, cucumber automation may encounter difficulties. Managing and maintaining a complete collection of Cucumber scenarios can be difficult as software system size and complexity rise. Future studies should investigate methods to deal with scalability problems, like enhanced test organization, modularization, and effective test execution mechanisms.

### 7.3 Integration with Test Management Tools:

Cucumber has its own reporting features, but it might be difficult to integrate it with traceability and test management solutions. The traceability of requirements, test coverage, and overall test reporting capabilities can all be improved by strengthening the interaction between Cucumber automation and test management systems.

#### 7.4 Support for Non-functional Testing:

Automation with cucumber mostly focuses on functional testing elements like user interactions and organizational behaviors. But non-functional testing aspects like performance, security, and usability can need more focus. Future studies can look into methods for efficiently extending Cucumber automation's capabilities to include non-functional testing elements.

#### 7.5 Industry-specific Customizations:

When using Cucumber automation, different sectors could have unique needs and problem domains that need to be solved. To meet the special requirements of diverse industries, such as banking, healthcare, or e-commerce, future research can concentrate on creating industry-specific best practices, guidelines, and customization possibilities.

#### 7.6 Integration with Development Processes:

Although Cucumber automation harmonizes business and testing practices, it can be quite important to integrate it into the broader software development process. Future studies can investigate ways to successfully integrate Cucumber automation with Agile and DevOps development approaches, providing efficient communication between developers, testers, and other stakeholders.

#### 7.7 Advanced Reporting and Analytics:

Although cucumber automation produces insightful test reports, additional developments in reporting and analytics can offer more in-depth perceptions into test outcomes and patterns. To improve the decision-making process and offer more thorough test insights, future research can concentrate on developing enhanced reporting and analytics capabilities, including visualizations, trend analysis, and predictive analytics.

### VIII. CONCLUSION

The study paper has, in its conclusion, emphasized the considerable impact and advantages of utilizing Cucumber automation in bridging the gap between business and testing practices. It is clear from a study of the results that Cucumber automation is essential for fostering cooperation, boosting productivity, and ensuring software quality.

Cucumber automation encourages good communication and mutual understanding between business stakeholders and testing teams by utilizing the Behavior-Driven Development (BDD) methodology. The scenario-based

testing methodology helps testers to confirm crucial business functionalities, improving test coverage and facilitating the early detection of defects. Cucumber automation's adaptability as a framework for test automation enables collaboration between developers and testers, producing reusable and maintainable automation code.

The study's findings highlighted how crucial it is to deal with issues including the steep learning curve, scalability, and integration with test management tools. Future study may further examine integration with development processes, industry-specific adaptations, and enhanced reporting and analytics capabilities.

Overall, the study finds that Cucumber automation is a potent tool for integrating business and testing procedures. Organizations can achieve effective software development processes, increased cooperation, and higher software quality by coordinating business goals with testing efforts. The research paper's findings offer insightful analysis and useful suggestions for companies wishing to use Cucumber automation in their software development initiatives.

Cucumber automation has the potential to revolutionize how companies and testing teams work together, leading to more successful software projects and enhanced stakeholder collaboration with additional developments and ongoing study in this field. By adopting Cucumber automation, organizations can move towards a unified strategy where business requirements are successfully converted into executable tests, ultimately resulting in higher-quality software products and customer satisfaction.

### REFERENCES

- [1]. Leotta, M., & Mauri, F. (2019). Leveraging Cucumber Automation for Business-Readable Acceptance Testing. In International Conference on Agile Software Development (pp. 223-239). Springer.
- [2]. Metzger, A., Kelter, U., & Pohl, K. (2019). A systematic mapping study on the combination of business process models and user interface descriptions. *Information and Software Technology*, 111, 99-115.
- [3]. Kaur, K., & Rani, A. (2020). A Systematic Review of Behavior-Driven Development Tools and Techniques. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(5), 1956-1962.

- [4]. Vandenryt, T., & De Smet, J. (2019). Cucumber to assess compliance of requirements. In Proceedings of the 27th IEEE International Requirements Engineering Conference Workshops (pp. 176-183). IEEE.
- [5]. Lombardo, C., Ceccarelli, A., & Matera, M. (2019). Towards Automatically Generated Test Reports in Cucumber. In International Conference on Web Engineering (pp. 583-594). Springer.
- [6]. Malyarenko, O., Tyrväinen, P., & Taibi, D. (2019). Empirical evaluation of Behavior Driven Development and automated acceptance testing. *Journal of Systems and Software*, 147, 201-218.
- [7]. Sahota, R. (2020). *The Agile Culture: Leading through Trust and Ownership*. Addison-Wesley Professional.
- [8]. Farahani, M. J., & Abdul Razak, S. (2019). Towards Improving Automated Test Design: A Systematic Literature Review. *Journal of Computer Science*, 15(7), 1053-1070.
- [9]. Zou, D., Ali, N., Ramzan, N., Iqbal, M. Z., & Ahmad, R. (2021). Evaluating BDD framework cucumber as a tool for automated testing: a systematic review. *Software Quality Journal*, 29(1), 177-224.
- [10]. Janzen, D. S., & Saiedian, H. (2018). Effective Use of Cucumber in an Undergraduate Software Engineering Course. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (pp. 766-771). ACM