

# Behavior-Driven Testing of Billing and Tax Modules: Applying Cucumber and Gherkin at Charter Communications

Udayan Verma

Denver, USA

[udayanverma7@gmail.com](mailto:udayanverma7@gmail.com)

## Abstract:

This report explores the application of Behavior-Driven Development using Cucumber and Gherkin to validate billing and tax modules at Charter Communications. By aligning technical and non-technical stakeholders, BDD enhanced compliance, accuracy, and maintainability of workflows. The study highlights methodology, case application, results, challenges, and recommendations for sustainable testing practices.

**Keywords:** Behavior-Driven Development, Cucumber Framework, Gherkin Syntax, Billing and Tax Compliance.

## I. Introduction

Charter Communications is an American telecommunications and mass media corporation that has millions of customers located in various regions. Billing and tax modules are at the foundation of financial operations in such a complex service environment and are the guarantee that customers receive the appropriate legal billing. Any inaccuracy in the billing or calculation of taxes can result in loss of revenue, consumer dissatisfaction, or legal action. Consequently, the precision, conformity and ease of maintenance of these workflows are critical. The traditional approach to automation can be quite useful but is likely to fail in establishing a common ground between non-technical and technical stakeholders. To overcome these difficulties, this report considers the deployment of Behavior-Driven Development through the Cucumber framework and Gherkin syntax. The goal is to show the benefits of BDD in collaboration, compliance, and testing streamlining.

## II. Background and Context

Tax and billing modules in large enterprises are necessarily difficult because of the combination of complex business rules, regional tax differences and ad hoc regulatory changes. Calculating the exact price to charge a customer depends on several factors. These consist of the plan chosen by the customer, any discounts or surcharges, and the local tax rates in the customer location [2]. An inaccurate calculation may cause inaccurate invoices, financial accounting, and regulatory breaches, all of which impose high risks on customer confidence and corporate image.

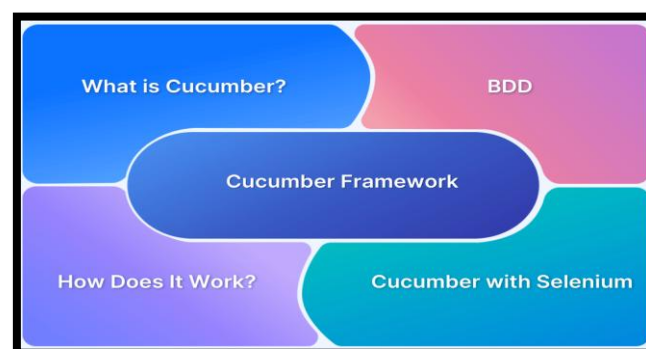


Figure 1: Cucumber Framework

The size of Charter Communications' business also exacerbates such issues because thousands of daily transactions must be handled with accuracy. The typical test automation frameworks have served well in regression testing yet fail to keep up with such requirements. Automated scripts are very technical, hard to maintain, and they are rarely comprehensible by business or legal stakeholders. This makes the intent of the requirements and the verification of results out of place. Behavior-Driven Development (BDD) emerged as a solution to constraints, emphasizing collaboration among developers, testers, and business reps based on natural language specifications. BDD makes requirements transparent and testable through business-readable test cases. It's useful in sectors with strict workflow compliance, like telecommunications and finance, linking technical work to business expectations.

### III. Methodology: BDD with Cucumber and Gherkin

Behavior-Driven Development is a software testing approach that involves developing test examples in the form of understandable business scenarios. Behavior-Driven Development (BDD) is a group-based software development procedure. It allows teams to describe system specifications in a clear, easy-to-read language. This language eases the interaction in technical and non-technical staff members [4]. This will ensure that the goal of the requirement is inevitably met in the development, testing, and validation of the requirement. One of the most used frameworks to apply BDD is the Cucumber framework. It can take test cases written in the Gherkin syntax, a structured natural language notation in the form of Given-When-Then. The context of action is indicated as given, then the operative matter being carried out is stated as When, and the result is indicated as Then. This template is clear for business analysts, QA engineers, developers, or legal officers. Scenarios are created together during sessions where requirements are modeled as specific scenarios. This collaboration reduces ambiguity and ensures all parties understand the business rules. For instance, a billing and tax situation may entail the following:

Scenario: Apply correct sales tax for New York customer  
 Given a customer located in New York with a monthly plan of \$100  
 When the invoice is generated  
 Then the system should apply 8.875% sales tax  
 And the total amount should be \$108.88

**Table 1: Billing and Tax Scenario**

The example below shows how actual business rules can be represented in a way that is executable as a test and clear. This way, traceability of requirements to automated tests ensures alignment, responsibility, and maintainability.

### IV. Application at Charter Communications

#### A. Context of Billing and Tax Workflows

Charter Communications deals with millions of customer accounts in various states with distinct billing requirements and regulations, along with the need to pay taxes. The billing system should be able to support recurrent charges, discounts, surcharges, and plan upgrades or downgrades, meeting the tax rate requirements imposed per federal, state, and local jurisdiction [5]. The consequences of errors occurring in these workflows are not only revenue loss, but also the danger of non-compliance, driving the need for rigorous and visible testing to be in place.

#### B. Introduction of BDD in 2022

In August 2022, Charter Communications implemented Behavior-Driven Development in using Cucumber and Gherkin to enhance the testing of billing and taxing sections [3]. This method was selected to help relate complicated financial regulations to test automation in a manner that can be verified by business and legal functions. BDD created cooperation between non-technical and technical teams, to get the guidelines accurately translated into scenarios that could be tested.

#### C. Collaboration Model

The BDD procedure involved integrating QA engineers, developers, business analysts and legal specialists in refining sessions. The needs were considered, explained, and transformed into Gherkin scenarios, so that

all the stakeholders involved could ensure that they are correct prior to being implemented. This model avoided having a force field on business rightness and technical implementation.

#### D. Example Use Cases

The billing accuracy was covered with scenarios to examine proper generation of bills, applying discounts, and upgrading plans. The tax compliance was driven by ensuring compliance with the state-specific taxation rates, exemptions and federal taxation. A typical Gherkin scenario would be:

Scenario: Exempt customer does not pay sales tax  
 Given a customer in Texas flagged as tax-exempt  
 When the invoice of \$200 is generated  
 Then the system should not apply state sales tax  
 And the total amount should remain \$200

**Table 2: Gherkin scenario**

#### E. Benefits Realized

This deployment enhanced mutual comprehension, helped identify inconsistency with compliance promptly, created less ambiguity in the requirement, and increased transparency and trustworthiness in financial processes [1].

### V. Results and Impact

A Behavior-Driven Development case study behind Charter Communications measured the returns on BDD billing and taxation. The coverage of the testing process was also enhanced, with readable business scenarios being used to cover a greater range of billing variants and taxation regulations that had been missed with traditional automation. The Gherkin syntax allowed expression of even the complex requirements of compliance in a way that was visible and executable. This plan guaranteed that our financial needs would be sufficiently and substantially tested [6]. A noteworthy boost in the care of stakeholders also took place. Legal departments, who only had some ideas about test automation, could now review and verify situations. This enhanced compliance assurance and minimized misinterpretation of regulations. Business analysts could conveniently trace requirements to test outcomes, which led to a seamless connection between business intent and validation. The method also decreased production defects from billing errors and tax miscalculations. Early stall detection in refinement sessions cuts later fixing costs. The test suite was easier to maintain, as Gherkin scenarios kept documentation aligned with requirements. Overall, BDD harmonized technology and business, boosting efficiency, compliance, accountability, and customer satisfaction in Charter's finance workflow.

### VI. Challenges and Limitations

Although BDD offered evident benefits, it caused issues in adoption. Teams not experienced with the methodology were on a steep learning curve because training and cultural accommodation were needed. Constant cooperation, which was good, became time-consuming. In addition to this, ill-constructed or inconsistent Gherkin scenarios threatened to diminish effectiveness, limit traceability and dilute the alignment between requirements and implementation.

### VII. Conclusion and Recommendations

The use of BDD, Cucumber, and Gherkin within Charter Communications resulted in a major efficiency gain in its billing and tax workflow validation. It allowed collaboration and transparency, which contributed to compliance and decreased errors. It is advisable to extend BDD to other areas where business requirements are complex and related to compliance, such as legal and regulatory requirements, trademarking, patenting, taxes, among others.

#### REFERENCES:

1. Deng, Y., Lou, G., Zheng, X., Zhang, T., Kim, M., Liu, H., Wang, C. and Chen, T.Y., 2021, June. BMT: Behavior driven development-based metamorphic testing for autonomous driving models. In *2021 IEEE/ACM 6th International Workshop on Metamorphic Testing (MET)* (pp. 32-36). IEEE.

2. R. Marchese, A. Zen, and Villafiorita, “GHERKIN\* AND CUCUMBER\* A NEW TEST CASE PATH COMPOSITION APPROACH TO TESTING RUBY ON RAILS WEB APPLICATIONS,” 2013. Available: [https://robzenn92.github.io/assets/thesis\\_zen\\_roberto.pdf](https://robzenn92.github.io/assets/thesis_zen_roberto.pdf)
3. Patkar, N., Chiş, A., Stulova, N. and Nierstrasz, O., 2021, October. Interactive behavior-driven development: a low-code perspective. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)* (pp. 128-137). IEEE.
4. [1]H. M. Sneed and C. Verhoef, “From COBOL to Business Rules—Extracting Business Rules from Legacy Code,” pp. 187–208, Jan. 2020, doi: [https://doi.org/10.1007/978-3-030-26574-8\\_14](https://doi.org/10.1007/978-3-030-26574-8_14).
5. R. K. Lenka, S. Kumar, and S. Mamgain, “Behavior Driven Development: Tools and Challenges,” 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Oct. 2018, doi: <https://doi.org/10.1109/icacccn.2018.8748595>.
6. H. M. Abushama, Hanaa Altigani Alassam, and F. 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), Feb. 2021, doi: <https://doi.org/10.1109/icceeee49695.2021.9429593>.