International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

|| Volume 8, Issue 7, July 2020 ||

Behaviour Driven Development Framework for Web Application Development

Vishnudevi T, Sathish Kumar G, Pavithra P

UG Student, Dept. of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

Assistant Professor, Dept. of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

UG Student, Dept.of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

ABSTRACT: Most of the time, customers request complex business logic to be implemented in software applications. Testing is the most relevant phase in SDLC. Testing is the quality assurance stage and done to point out the errors and bugs came during development so that it gives satisfaction and reliability to the customer over application. Therefore, as long as business requirements grow, the pressure increases on the testing team to deliver the product with high quality in a very tight time. Manual testing is not suitable for critical and complex applications in terms of both human resources and time. Therefore, there is a strong need to propose an automated testing framework which could reduce the overall software testing time. Automation testing has been introduced to overcome manual testing problems. In this article. I have used the approach of BDD(Behaviour Driven Development) framework called specflow in which specifications are written in the commonly used quasi-natural language Gherkin and demonstrate its applicability on IMDB movie web application.

KEYWORDS: Testing, Behaviour Driven Development (BDD), Specflow,Gherkin, Cucumber.

I.INTRODUCTION

In typical commercial contexts, modern software engineering is characterised by a strong customer focus and, as a result, fluctuating requirements. These processes start off by gathering requirements from the customer in the form of case scenarios and subsequently development is built around these scenarios culminating in their implementation and demonstration to the customer.

This process, better known as behaviour-driven development,re-orientates the development process so as to ground all work around business level customer specifications. Once the customer scenarios are implemented, the process tends to take on an iterative form whereby the customer is again asked to review the specification and possibly introduce further scenarios. To keep the specification language as simple as possible for non-technical customers, scenarios are typically specified in terms of a quasi-natural language called Gherkin .

In order to automate test execution, clauses from the Gherkin specification are associated with executable code using a technology such as Cucumber(Specflow). This executable code effectively turns the specification into an automated acceptance test which is used to test that the system adheres to the customer's specifications. Whilst the automation achieved through the running of Gherkin specifications has significantly shortened the test execution process and made it repeatable.

Matt the lead developer on Cucumber-Ruby,the open source tool for running executable specifications that help bridge the communication gap betweenbusiness stakeholders, testers and developers. He's co-author of The Cucumber Book with Aslak Hellesøy, and Cucumber Recipes with Ian Dees and AslakHellesøy, and The Cucumber for Java Book with Seb Rose and AslakHellesøy[1].David Chelimsky was responsible for developing the RSpec-Rails which facilitated the integration with Ruby on Rails. The mocking for the resources has been developed by them[2] .Monier and El-mahdy[3] have made a publication on evalution of automation testing tools.They conducted several analysis of web testing tools and classified them to two types.

Comparison among the two types was carried out and this help to choose the testing based on user requirements. Cucumber is a tool that supports Behaviour-Driven Development (BDD) - a software development process that aims to enhance software quality and reduce maintenance costs.Cucumber executes executable specifications written in plain language and produces reports indicating whether the software behaves according to the specification or not.Cucumber reduces the effort to keep requirements specifications, tests and documentation in sync - with Cucumber they are all the same documents - a single source of truth for everyone on the team.The basics of the software framework was given in the wiki[4].

Cucumber recepies book defines different tools that can be combined to automate the behaviour in many situations. Each recipe is short, elegant and precisely to the point. The book is focused on using Ruby whenever it is possible. There are other recipes using Java and .net are available,all recipes are translatable to other Cucumber implementations[5].

II.LITERATURE REVIEW

Dietmar Winkler along with the colleagues published the paper which focuses on the challenges and requirements for flexible test automation in the automation systems and production systems domain .Along with it they proposed the flexible test automation framework (TAF), presented a pilot implementation of the TAF with selected tools, and conceptually evaluated strength.Their results shows the feasibility of the testing approach in the evaluation use case to improve test automation processes in the automation systems domain. Although the implementation of the TAF supports efficient configuration and execution of test processes, additional effort is required to initially setup the test automation framework[6].

Milad Hanna along with his colleagues published the paper which has the goal topropose another programming computerized testing system which gives more assistance for analyzers in the general programming computerization testing process. Their proposed structure can effectively be utilized for computerizing the test contents creation process. Their proposed approach also generates the test cases into a standard tabular format which is easier and stricter than writing them manually in normal English language[7].

Christian Colombo along with his colleagues proposed a methodology for changing over business level determinations composed utilizing Given-When-Then common language documentation into models that can in this manner be utilized to create and test projects. This strategy can work off natural language specifications which adhere to three simple conventions and if applied in a company where automated testing is already the norm, model creation and consequent experiment age can be accomplished at insignificant expense[8].

SrashtiLariya along with her colleagues published paper in which she used BDD Framework with selenium web Driver.In her publication brings out the importance of BDD such as to improve cooperation and understanding between including parties,it likewise gives high perceivability to all members as it utilizes universal language which is understandable by all. Because of this, Developers can give excellent item which addresses client issue correctly and diminish the upkeep cost for code[9].

Carlos Solís and Xiaofeng Wang published a paper shows that the majortools come up short on the help of the BDD qualities identified with the planning and testing stages. In this way one future investigation could expand a current BDD toolbox or build up another one dependent on the proposed applied model. The new toolbox will offer help to the product advancement exercises that need joint effort among business and improvement group[10].

Abigail Egbreghts on his paper identified the concepts of BDD in literature. From this survey, 6 primary ideas were distinguished; pervasive language, prerequisites detail, acknowledgment tests, devices, cooperation, and robotization. These discoveries improve our comprehension of BDD ideas and its practices in writing. The present investigation affirms past discoveries from Solis and Wang about the ideas of BDD and contributes with extra proof that recommends in paper, BDD isn't constantly actualized all in all light-footed strategy[11].

GarmLucassen along with the colleagues published the paper which proposed Behaviour-Driven Traceability technique (BDT) that, expanding on the coordinated programming improvement process Behaviour-Driven Development, endeavors to build up pervasive discernibility .Starting testing of our model on the open source venture Archive of Own produces promising yield that persuades us to keep researching this area.Both crude and standardized BDT yield have merit: the previous catches the pervasiveness and significance of few classes and strategies, while the last features classes and techniques that are imperative to a single user story[12].

Thiago Rocha Silvapublished the paper which verify potential problems and inconsistencies when working with multiple design options and complex task models. We are also developing a tool to support the creation, visualization and execution of the tests[13].

PreetiKhandokar in her paper says key is to achieve high quality product and test automation in the same sprint is reduce the complexity of test automation and making it increasingly helpful for end customer or the user. It gives early and persistent input to the spry application improvement. By creating test automation at a beginning time achieved higher quality product. It diminishes 40-60 percent exertion for mechanized test content age over manual testing[14].

Niranjanamurthy along with his students published the paper which says ,Today everything is web based in this way it is turning out to be increasingly complex. For this, tremendous data stage, quick release cycle as well as quick regeneration is required. This requires the web application to be exhaustive, expansibility and effectiveness. For this some structure additionally incorporates automation tools. You can diminish the expense caused for authorizing utilizing QTP Using Selenium as the Functional Test Automation Tool[15].

III.PROPOSED METHODOLOGY

3.1.TECHNOLOGY DESCRIPTION

Behavioural driven development methodology is used for software development where all the member of team communicates to understand the entire requirement. Here product owner, tester, business analyst and developer talk about business needs and collaborate around the requirement. The owner specifies its need and behaviour they want to see in software. These interactions help developer and tester to deliver the well-defined result. With the help of user stories, they get clear understanding and write the requirements into feature file which is in plain English language known as gherkin. Some advantages of BDD

1) Strong collaboration 2) High visibility 3) The software design follows business value 4) ubiquitous language 5) More confidence from the developers' side

6) Lower costs

3.1.1Specflow

Specflow is software testing tool, it builds understanding between the business manager and software developer, specflow is created to process Behavioural driven development (BDD) in which requirements are implicated in a feature files which is in plain English called as Gherkin language. It is in the format of “Given, When, Then”. The test scenarios written in Gherkin language are placed in a Feature File.

3.1.2 Feature File

When test cases are written in files then these file are known as feature file. In cucumber, for every functionality we prepare different feature files. For specific feature file we create a scenario and scenario outline. File is in “.feature”are

1) Feature –It defines specific functionality in test case.

2) Scenario –It defines how the functionality work.

3) Given –Pre-requisites.

4) When –The action to be done.

5) Then – Validation/Verification.

Scenario outline - we will give multiple inputs for a single scenario so that it can check multiple input.

3.1.3 Step Definition

To execute our job written in feature file, we call an intermediate step definition class. All the steps of feature file are mapped with the code of function for execution. Now specflow execute all the scenario of feature file by scanning step definition file

3.2 IMDB MOVIE APPLICATION

Imdb movie application is an web application which provides the user with list of movies along with cast details,producer details ,year of release.The worker will fetch the list of movies from the external apiImdb website and store it in the database.The Repository class will do the database operations.The Service class contains the business logic.The Controller will connect the Asp.net application to the VueJs which holds the front end design for the Imdb movie application.User can able to see list of movies details,list of actor details,list of producer details.InImdb movie application,the following functionalities are tested:

1.The test for fetching data from external API.

2.The test for CRUD operations of movie.

3.The test for CRUD operations of actor.

4.The test for CRUD operations of producer.

3.3 Working Flow of BDD:

The first step is collecting user requirements.Once if we clarity on the requirements, need to come up with the scenario for user requirements, then need to implement the scenarios using step definition. Also Mocks of repository or API endpoints can be created using MOQ. The mocks helps to successfully run the testcase even in case of network issues. Once the tests were successful ,The code of user requirements can be developed. So this process helps to test the user requirements first and once the requirements are tested then start with code.This method is called as code first approach. So any changes in the requirements won't affect the developers code very much,only test cases gets changed.

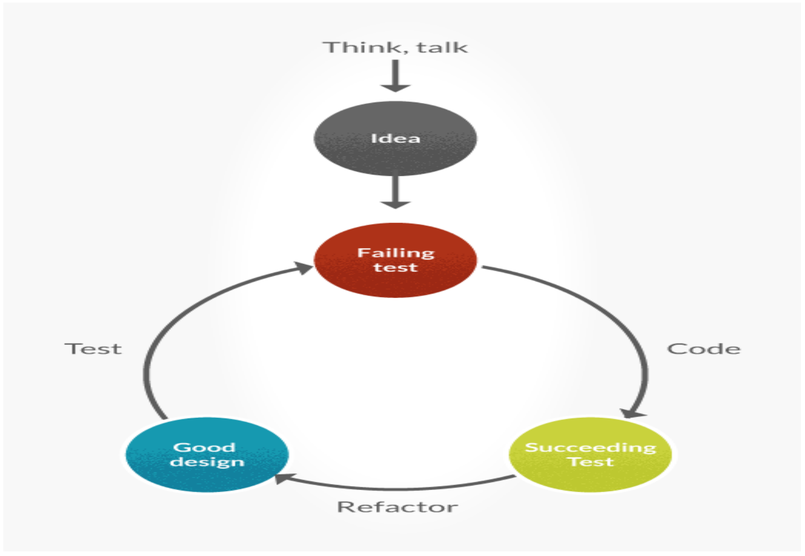


FIG 1: BDD WORK FLOW

IV.RESULTS AND DISCUSSION

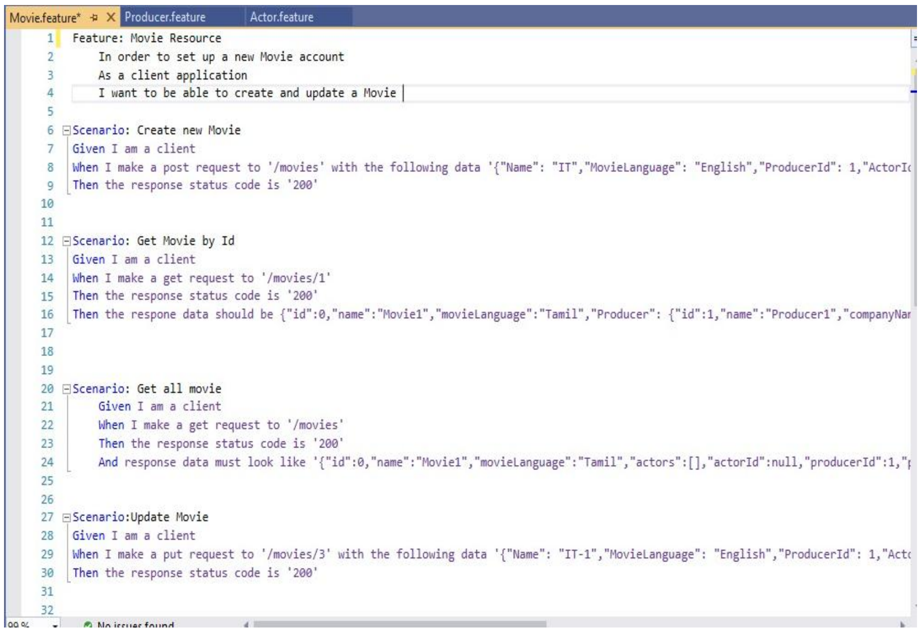
The aim of this project was to show the uses of BDD.The way this is different from other web application is,it involves test first approach.In this movie app the behaviour of the user requirement is verified and the implementation process was carried out.So several changes after implementation is reduced,which leads to saving of time.

4.1 Feature File(Movie –CRUD operations):

The behaviour of the user requirement was written in the gherkin language.This file is named as feature file.In this feature file CRUD behaviour of the movie is tested.For example, requirements are movie needs to be restored from the database(GET),the movie needs to be created,the movie can be edited and the movie can be deleted.Each of these are individual behaviour,so a scenario is written for each behaviour and tested.

4.2 FEATURE FILE(Worker):

In this feature file, The fetching of movie from external API behaviour was tested. This is the feature file for testing the behaviour of the worker. The Worker fetch the list of movies from the external API. In case of getting data from external API ,we need to test whether the response matches our data, or else needs to convert the response to required data and also needs to check the pages of the response, the query used for filtering. All these are tested using BDD.

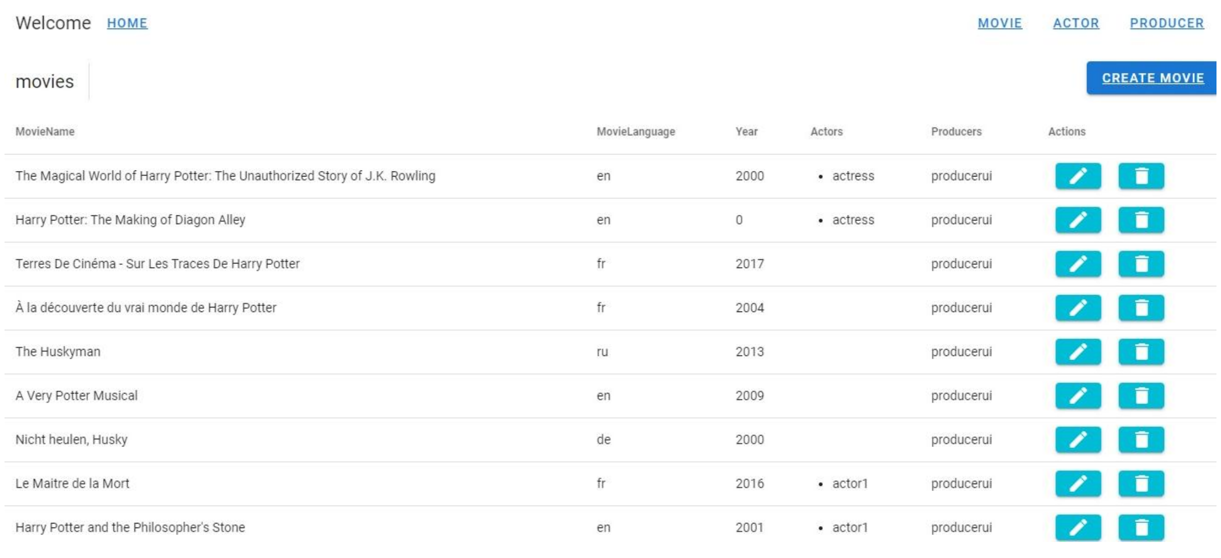


```
Feature: Movie Resource
    In order to set up a new Movie account
    As a client application
    I want to be able to create and update a Movie

Scenario: Create new Movie
Given I am a client
When I make a post request to '/movies' with the following data '{"Name": "IT","MovieLanguage": "English","ProducerId": 1,"ActorI
Then the response status code is '200'


Scenario: Get Movie by Id
Given I am a client
When I make a get request to '/movies/1'
Then the response status code is '200'
Then the respone data should be {"id":0,"name":"Movie1","movieLanguage":"Tamil","Producer": {"id":1,"name":"Producer1","companyNa



Scenario: Get all movie
    Given I am a client
    When I make a get request to '/movies'
    Then the response status code is '200'
    And response data must look like '{"id":0,"name":"Movie1","movieLanguage":"Tamil","actors":[],"actorId":null,"producerId":1,"


Scenario:Update Movie
Given I am a client
When I make a put request to '/movies/3' with the following data '{"Name": "IT-1","MovieLanguage": "English","ProducerId": 1,"Act
Then the response status code is '200'
```

FIG 2: Movie feature file

4.3 FRONT END:

The front end was designed using VueJs.In the front end the CRUD operations of movie ,actor and producer were implemented.Once the tests are passing,frond end can be implemented.Here the front end is implemented using vuejs. The home page contains tab for movie ,actor and producer.The movie tab supports creating of movie,edit of movie and delete of movie.Similarly actor tab supports adding of actor,editing of actorand deleting of actor.The producer tab supports adding of producer,editing of producerand deleting of producer.



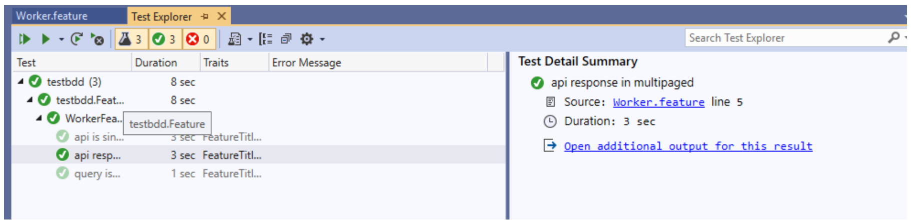
movies

MovieName	MovieLanguage	Year	Actors	Producers	Actions
The Magical World of Harry Potter: The Unauthorized Story of J.K. Rowling	en	2000	• actress	producerui	✎ 🗑
Harry Potter: The Making of Diagon Alley	en	0	• actress	producerui	✎ 🗑
Terres De Cinéma - Sur Les Traces De Harry Potter	fr	2017		producerui	✎ 🗑
À la découverte du vrai monde de Harry Potter	fr	2004		producerui	✎ 🗑
The Huskyman	ru	2013		producerui	✎ 🗑
A Very Potter Musical	en	2009		producerui	✎ 🗑
Nicht heulen, Husky	de	2000		producerui	✎ 🗑
Le Maitre de la Mort	fr	2016	• actor1	producerui	✎ 🗑
Harry Potter and the Philosopher's Stone	en	2001	• actor1	producerui	✎ 🗑

FIG 3:Web App

4.4 TESTS:

The test explorer contains tests which we were written.Each scenario gets converted to test.It is used to evaluate the user behaviour. Once the tests are run,it tests each line of the scenario.During the test run ,it evalulate the implementation of the test scenario.The green colour shows that all user behaviour was tested and it was working as expected.The failing of the tests shows there is some error in terms of implementation of user behaviour or the user behaviour has been changed.



Test	Duration	Traits	Error Message
testbdd (3)	8 sec		
testbdd.Feat...	8 sec		
WorkerFea...			
api is sin...	3 sec	FeatureTitl...	
api resp...	3 sec	FeatureTitl...	
query is...	1 sec	FeatureTitl...	

Test Detail Summary

api response in multipaged

Source: Worker.feature line 5

Duration: 3 sec

Open additional output for this result

FIG 4: Test Explorer

The BDD approach helps in validation of user requirements beforehand of code implementation.This helps in time saving.Forex,If an user requirement is changed after implementation,it is time consuming,this can be avoided by BDD first approach.TDD focus on implementation of functionality whereas BDD focus on behaviour of the application.BDD is readable and written in gherkin language whereas TDD uses programming languages which is not understandable for an end user.

V.CONCLUSION

The key is to achieve high quality product and test automation in the same sprint.BDD approach is used to reduce the complexity of test automation and making it more useful for end client. It leads automation at every level to achieve rapid development. It supports early automation and early life cycle validation and single-click generation and execution of automation scripts. Developing test automation at an early stage have achieved higher quality product at every stage of product development. It reduces 40-60 percent effort for automated test script generation over manual testing. It ensures high defect detection rates (95-98%) due to high test coverage.

REFERENCES

[1] Wynne M., Hellesoy A., Tooke S.: “The cucumber book: behaviourdriven development for testers and developers”, Pragmatic Bookshelf, 2017.

[2] David Chelimsky, Dave Astels, Bryan Helmkamp, Dan North, Zach Dennis &AslakHellesoy (2010): The RSpec Book: Behaviour Driven Development with Rspec, Cucumber, and Friends (The Facets of Ruby Series). Pragmatic Bookshelf

[3] Ian Dees, Matt Wynne &AslakHellesoy (2013): Cucumber Recipes: Automate Anything with BDD Tools and Techniques. Pragmatic Bookshelf.

[4] Gherkin: Gherkin Wiki. Available at http://github.com/cucumber/cucumber/wiki/Gherkin.

[5]M. Monier and M. M. El-mahdy, "Evaluation of Automated Web Testing Tools," International Journal of Computer Applications Technology and Research, vol. 4, no. 5, pp. 405-408, 2015.

[6] Dietmar Winkler , KristofMeixner, Stefan Biffl “Towards Flexible and Automated Testing in Production Systems Engineering Projects”, Christian-Doppler-Laboratory for Security and Quality Improvement in the Production System Lifecycle TechnischeUniversität Wien Vienna, Austria

[7]Milad Hanna, AmalElsayedAboutabl ,Mostafa-Sami M. Mostafa “Automated Software Testing Framework for Web Applications”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 9758-9767 © Research India Publications.

[8]Christian Colombo ,MarkMicallef. Mark Scerri,” Verifying Web Applications: From Business Level Specifications to Automated Model-Based Testing”, PEST Research Lab Department of Computer Science University of Malta, Malta.

[9] SrashtiLariya, Dr. Sameer Shrivastava, Er. SumitNema,” Automation Testing using Selenium Web Driver & Behaviour Driven Development (BDD)”, Department of Computer Science &Engineering ,Global Nature Care Sanghtan’s Group of Institution, Jabalpur, India.

[10]Carlos Solís,XiaofengWang.”A Study of the Characteristics of Behaviour Driven Development”,Lero, the Irish Software Engineering Research Centre University of Limerick Limerick, Ireland.

[11]Abigail Egbreghts,”A Literature Review of Behaviour Driven Development using Grounded Theory”,University of Twente P.O. Box 217, 7500AE Enschede the Netherlands.

[12]GarmLucassen, FabianoDalpiaz, Jan Martijn E.M. van der Werf, SjaakBrinkkemper and DidarZowghi,"Behaviour-Driven Requirements Traceability via Automated Acceptance Tests",University of Technology Sydney, Australia.

[13]Thiago Rocha Silva,"Definition of a behaviour-driven model for requirements specification and testing of interactive systems",Université Paul Sabatier – Toulouse III Toulouse, France.

[14]PreetiKhandokar,"Advantages OF BDD Testing",Datamatics Global Solutions Ltd,software testing conference,17th annual international conference,2017,Bangalore.

[15]Niranjanamurthy M, Arun Kumar R, SahanaSrinivas, Manoj RK,"Research Study on Web Application Testing using Selenium Testing Framework",Assistant Professor, Department of MCA, MSRIT, Bangalore-54, INDIA,2014.