

A Hybrid of Automated and Manual Testing in an Agile Environment

[RosnisaAbdullRazak | Muhammad Dhiauddin Mohamed Suffian]

Abstract— Software testing refers to the process of verifying as well as validating the successful functioning of particular software. Manual and Automated testing are example of testing that is adopted in an Agile environment. There are many misconceptions in an agile environment regarding both manual and automated testing. Some practitioners believe that automated testing is the bee's knees and exists as a replacement for manual testing while others believe that manual testing is a simple set of step-by-step tasks that anyone can run through to check an expected output. Therefore, this belief has created a divide in the testing community between manual and automated testing. In this paper, we will discuss more on combination of both manual and automated testing in Agile Environment. A case study shows that the combination yields better results.

Keywords—Software Testing, Manual Testing, Automated Testing, and Agile Environment.

I. Introduction

Software testing is the process of executing a program or system with the intent of finding error [1]. It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required result [2]. Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical system fails in fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible [3].

Software bugs will always exist in any software module with moderate size, not because programmers are careless or irresponsible, but because the complexity of software and human have only limited ability to manage complexity. It is also true that for any complex systems, design defects can never be completely ruled out.

Rosnisa Abdull Razak(*Author*)
Product Quality and Reliability Engineering
MimosBerhad
Malaysia

Muhammad Dhiauddin Mohamed Suffian(*Co-Author*)
Faculty of Computing
UniversitiTeknologi Malaysia
Malaysia

In the world of software development, agile typically refers to any approach to project management that strives to unite

teams around the principles of collaboration, flexibility, simplicity, transparency, and responsiveness to feedback throughout the entire process of developing a new program or product.

In a waterfall (traditional) approach, any given project's workflow proceeds in a linear series of sequential steps, progressing down the chain of production from *Requirements>Design>Implementation>Verifications>Maintenance*. This split between software developers and software testers, positioning them as separate entities at different points.

The organization of this paper is as follows. Section II describes the prior and existing works related to this study. In section III, it discusses the background of the manual and automated testing while in section IV; it describes the hybrid manual and automated testing in more detail. Finally, we conclude our study with the recommendation for future works in section V.

II. Related Work

It cannot be denied that agile has gained popularity due to the perception that it could produce software that meet user needs while at the same time incorporated high quality attributes. Agile has been understood, implemented and practiced in variety of forms either as solution, validation, opinion, philosophical, experience or evaluation purpose [4]. Most of the time, agile is always associated with test-driven development (TDD), automated testing, acceptance testing and unit testing [5]. The introduction of agile has always been the way to reduce manual efforts in executing the test. This can be seen how Scrum as one of the approach in agile was used to plan and organize manual test efforts through backlogs prioritization, investigation and exploration, burn-down charts, test effort prioritization and backlog summary [6]. This has led to the effort on merging the agile and traditional software development process to meet the changing needs of software [7]. This attempt which was considered as extension to Boehm and Turner's work relies on following core configuration process: profiling, aligning, preparing, running and checking. This demonstrates how the process of developing any software should be more dynamic and flexible.

Agile puts strong emphasis on the need for automation including test automation. There are various strategies when automating the tests: both automation of unit and system test by testers, automation of unit test by developers while automation of system test by testers, or both testers and

developers collaborate to automate unit and system test [8]. However, risks associated with implementing the test automation initiative should also be taken into consideration to ensure its success. [9] Stressed that collaboration becomes the key for everything in any agile project. Besides that, suitable testing tools should match with the test strategy and selected agile method. For any acceptance test in each sprint, only automate each software layer when there is a need to do that. Simple approach should be adopted when automating the tests and this must practice by every team member, not only testers. In compliance of the testing principle to have early testing as possible [10], automation in agile to test security and performance should be done early as well. It must not also be forgotten to document and get feedback by using test automation. Selenium, JUnit, PHPUnit, TestLink and Mantis are among popular testing tools that have been used to materialize this implementation [8][9][11][12].

III. Manual and Automated Testing in an Agile Environment

There are seems to be a divide in the testing community between manual and automated testing. Software can be tested either manually or automatically. The two approaches are complementary: automated testing can perform a large number of tests in little time, whereas manual testing uses the knowledge of the testing engineer to target testing to the parts of the system that are assumed to be more error-prone.

Some managers think automated testing is *extra* work. And since they have projects to deliver, there is no time to be fooling around. Somehow they overlook the fact that most project risk and uncertainty is the bug fixing part before the product is ready to be deployed.

In this section, we will discuss when to automate and when to manually test. We will also discuss pros and cons of automated and manual testing.

A. Manual Testing

Manual testing is the process of manually testing software for defects. It requires a tester to play role of an end user, and use most of features of the application to ensure correct behavior. To ensure completeness of testing, the tester often follows a written test plan that leads them through a set of important test cases [13].

The manual testing is very basic type of testing which helps to find the bugs in the application under test. The main goal of manual testing is to make sure that the application under test is defect free and software application is working as per the requirement specification document.

Manual testing is a method used by software developers to run tests manually. There are many manual testing types which are carried out manually as well as automatically. The software testing consists of two type of testing:

- **Functional Testing**
To check the software is working as per the functional requirement specification. This type of testing contains four front-end testing (GUI, Control flow, Input Domain, and Output or manipulation) and one back-end testing (Database testing).
- **Non-Functional Testing**
To check the software is working as per the non-functional requirement, which contains characteristic of the software to be developed like Usability, Compatibility, Performance and Security.

B. Automated Testing

In software testing, test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to performance manually. [14]

Automated testing uses automation tools to write and execute test cases; no manual involvement is required while executing an automated test suite. Usually, testers write test scripts and test cases using the automation tool and then group into test suites. The main goal of automated testing is to increase the test efficiency and develop software value.

C. Automated vs Manual Testing – pros and cons [15]

TABLE 1
 AUTOMATED VS. MANUAL TESTING PROS AND CONS

| Testing | Automated Testing | Manual Testing |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pros | <ul style="list-style-type: none"> • Run a set of tests repeatedly • Ability to run automation against code that frequently changes • Run automation scenarios to catch regression • Testing a large test matrix | <ul style="list-style-type: none"> • Test case only runs twice. Less cost than automate it. • Allows tester to perform more ad-hoc |
| cons | <ul style="list-style-type: none"> • It costs more to automate. • Can't automate visual reference (color) | <ul style="list-style-type: none"> • Time consuming • Each time there is a new build, tester must rerun all the test cases |

IV. A Hybrid of Manual and Automated Testing in an Agile Environment

As describes earlier, this study is to evaluate the successfulness of a hybrid manual and automated testing. A testing strategy can be manual or automated. With a manual strategy, the more traditional approach, testers prepare test cases based on the functional requirement. An automated testing strategy tries to remove the tediousness of the process by relying on a software tool that generates test scripts from the functional requirements.

Automated and manual strategies are often thought of as completely distinct, and usually supported by different tools. In fact they are complementary, since each has weaknesses that the other addresses.

In this section we will discuss more on the key principles from scrum and related agile methodologies and how we integrate manual and automated testing in during sprint execution stage.

A. Release Planning

The purpose of release planning is to establish a release strategy that the Scrum Teams and the rest of the organizations can understand and communicate, which lays out the overall project[16].It is important for technical people to make the technical decision and business people to make the business decision.

During this stage, scrum team member will obtain functional and non-functional requirements and transfer to product backlog and establish test strategy (what and when). The functional and non-functional requirements then will be created as a User Story in JIRA [17].

B. Story Elaboration (Sprint N-1)

The purpose of story elaboration is to establish sprint goal, select and elaborate user stories in JIRA for upcoming sprint. Scrum team member will clarify and update product backlog, determine sprint goal, select backlog to be implemented in the sprint and detail out the user story.

C. Sprint Planning (Sprint N)

The sprint planning is time boxed to 2-4 hours, and shall be attended by scrum team member. During the meeting the team will break user story into tasks in JIRA. Then they will estimate the tasks effort. The scrum team defines a sprint exit criteria (known as Definition of Done) which signifies the team's commitment and discipline to deliver.

D. Sprint Execution (Sprint N)[21]

The purpose of sprint execution is to implement the selected user stories by executing sprint backlog and validate

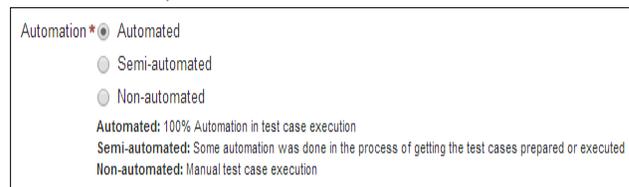
it against sprint goal. During the execution of a sprint cycle, the project team gathers every day in a short daily sprint cycle meeting (for a 15 minutes), where each team member shares their daily updates. In this section we will discuss more on creating test cases and then identifying and selecting the test cases for manual or automated testing.

In this section, we will explain more on how to create manual test cases, automation test scripts and how to execute manual and automated test cases in Sprint Execution phase. We create manual test cases in JIRA, then create automated test scripts using tool PyTestFramework and execute the test cases manually or run automated test script as part of regression testing.

Scenario 1: Create a Test Case in JIRA

Task A – Create test case for FEAT XYZ. The test case creation is complete when the following activities are fulfilling, otherwise this task still in-progress state:

1. Identifying and selecting test case to be automated, semi-automated, or non-automated.



The method of identifying and selecting test cases is depending on your application:

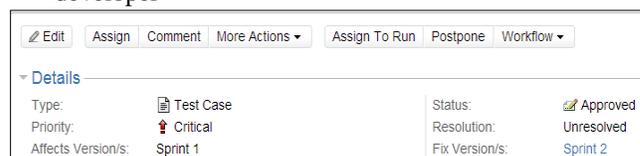
- Test case executed with different set of data
- Test case executed with different browser
- Test case executed with different environment
- Test case executed with complex business logic
- Test case executed with different set of users
- Test case involves large amount of data
- Test case has any dependency
- Test case required special data

If the tester chooses non-automated then the tester shall executed the test case manually. Manual testing is involves an exploratory testing, scenario-based and ad-hoc testing.

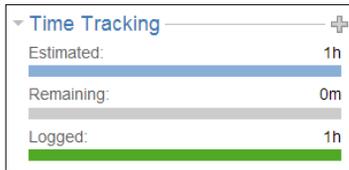
If the tester chooses semi-automated then the tester shall executed the test case manually and some automated testing.

If the tester chooses automated then the tester can start preparing the test scripts and perform the automated testing. Refer to scenario 2 for test scripts creation.

2. The test cases has been verified and approved by developer



3. Estimation for creating test cases has been logged



Scenario 2: Create automated Test Script using PyTestFramework[18]

Task B - Create automated test script for FEAT XYZ. The test script creation is complete when the following activities are fulfilling, otherwise this task still in-progress state:

1. Setup the environment

1. You need selenium server; hub and node are properly setup before running the tests. Please refer to [Selenium WIKI](#)

2. Example on hub, run

```
java -jar selenium-server-standalone-2.5.0.jar-role hub
```

3. Example on node to connect to hub ,run

```
java -jar selenium-server-standalone-2.5.0.jar-role  
webdriver-hub http://<HUB_NAME/IP>:4444/grid/register -  
port <PORTNO>
```

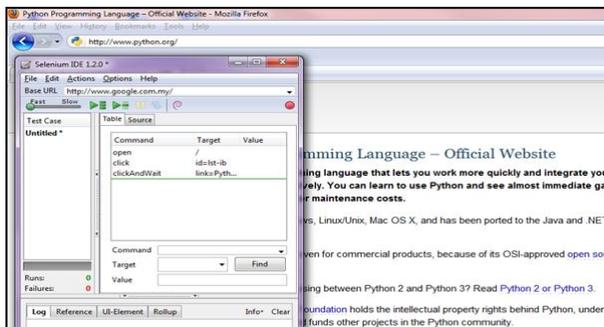
i.e.

```
java -jar selenium-server-standalone-2.5.0.jar-role  
webdriver-hub http://localhost:4444/grid/register -port 5555
```

4. You can have as many nodes as you want as per selenium grid features.

5. Record the test case using Selenium IDE [23]

1. Open Firefox web browser and go to Google
2. Click on Tools > Selenium IDE.
3. Type anything (eg. Python) to record the test steps.
4. Click on python.org link.
5. Click on Red button to stop recording.



6. Export the recording from Selenium to python test script and do some code modification to make it works in python environment.

1. Import functionR
2. Modify def Setup and add function.setUp (self)
3. Edit the base URL

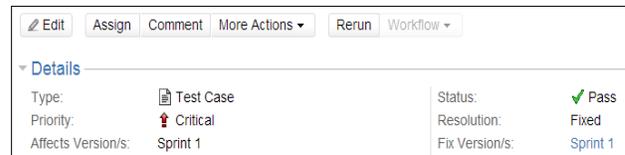
```

1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.common.exceptions import NoSuchElementException
4 import unittest, time, re
5 #You need to import functionR.py file as at least setUp function is located there.
6
7 import functionR
8 #Specify your pre-test case title as part of the class
9
10 class Anyname(unittest.TestCase):
11     def setUp(self):
12         #This is where setUp function from function.py is called
13         functionR.setUp(self)
14
15     #Your Selenium2.0/webdriver Python test cases should start here. Below is the example
16
17     def test_anyname(self):
18         driver = self.driver
19         #Enter the base URL
20         driver.get("http://www.google.com.my/")
21         driver.find_element_by_id("list-ib").click()
22         driver.find_element_by_link_text("Python Programming Language - Official Website").click()
23
24     def is_element_present(self, how, what):
25         try: self.driver.find_element(by=how, value=what)
26         except NoSuchElementException, e: return False
27         return True
28
29     def tearDown(self):
30         self.driver.quit()
31         self.assertEqual([], self.driver.errors)
32
33 if __name__ == "__main__":
34     unittest.main()
35
    
```

Scenario 3: Execute manual Test Cases and automated Test Scripts

Task C – Execute manual test cases or automated test scripts for FEAT XYZ. The test execution is complete when the following activities are fulfilling, otherwise this task still in – progress state:

1. All related test cases have been executed and passed in JIRA



Manual testing execution can be done by using exploratory or scenario based testing. Tester will attached the automated test scripts into the test case and update the Test case status to "Pass" once the testing is completed.

2. Test logs or test reports are completed

If the tester chooses semi or 100% automated testing, then the tester will attached the test scripts into test cases in JIRA. Refer below on how to run automated test scripts in Jenkins environment.

1. Run the python test script

```
Python Runx.py
```

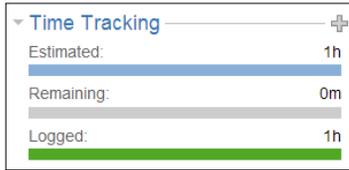
2. Integrate with Jenkins to run batch test scripts and to run overnight as part of regression testing. Below figure is the Build regression history, red light is displayed if the build failed due to bug or issue found during automated regression test.



3. Bug on failed test cases has been raised in JIRA.



4. Estimation on executing test cases has been logged in JIRA



E. Sprint Review

The sprint review meeting is time boxed to 2 hours. The purpose of the sprint review is for the team to present to the product owner the functionality that is done.

F. Sprint Retrospective

After the sprint review and prior to next sprint planning meeting, the scrum team has a retrospective meeting. This is final activity in sprint that indicates the sprint has been completed.

v. Conclusion

Testing of software can be done in both automation and manual testing method, and it's totally depends on the project requirement, budget associated with the project, and which testing method will be benefited to the project.

This paper presented the experience in performing manual and automated testing in agile software development Scrum. These experiences showed that it is feasible to do both automated and manual testing during test execution cycle. To execute the test cases first time using manual testing will be very much useful. But it is not sure that it will catch the regression defects under frequently changing requirements. Automation testing is very much helpful regressions in testing where code changes frequently.

Acknowledgment

The authors wish to thank the other members of the Product Quality and Reliability Engineering (PQRE) groups at MimosBerhad for their help throughout the data collection in this paper.

References

[1] Myers, Glenford J., *The art of software testing*, Publication info: New York: Wiley, c1979. ISBN: 0471043281.

[2] Hetzel, William C, *The Complete Guide to Software Testing, 2nd ed.* Publication info: Wellesley, Mass.: QED Information Sciences, 1988. ISBN: 0894352423

[3] http://www.rstcorp.com/definitions/software_testing.html

[4] T.D. Hellmann, A. Sharma, J. Ferreira and F. Maurer, "Agile Testing: Past, Present, and Future - Charting a Systematic Map of Testing in Agile Software Development", Agile Conference (AGILE) 2012, pp.55-63, 13-17 August 2012.

[5] T.D. Hellmann, A. Chokshi, Z.S.H. Abad, S.Pratte, F. Maurer, "Agile Testing: A Systematic Mapping across Three Conferences: Understanding Agile Testing in the XP/Agile Universe, Agile, and XP Conferences", Agile Conference (AGILE) 2013 , pp.32-41, 5-9 August 2013.

[6] A. Geras, "Leading Manual Test Efforts with Agile Methods", Agile Conference (AGILE) 2008, pp.245-251, 4-8 August 2008.

[7] A. Geras, M. Smith and J. Miller, "Configuring Hybrid Agile-Traditional Software Processes", Book Section on Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science, Berlin, Springer, pp.104-113, 2006.

[8] E.Collins, A. Dias-Neto and V.F. de Lucena Jr., "Strategies for Agile Software Testing Automation: An Industrial Experience", 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp.440-445, 16-20 July 2012.

[9] E.F. Collins and V.F. de Lucena Jr., "Software Test Automation Practices in Agile Development Environment: An Industry Experience Report", 2012 7th International Workshop on Automation of Software Test (AST), pp.57-63, 2-3 June 2012.

[10] R. Black, E. van Veenendaal and D. Graham, "Foundations of Software Testing ISTQB Certification – 3rd Edition", Cengage Learning, EMEA, 2012.

[11] A. Holmes and M. Kellogg, "Automating Functional Tests Using Selenium", Agile Conference (AGILE) 2006, pp.271-275, 23-28 July 2006.

[12] R.A. Razak and F.R. Fahrurazi, "'Agile testing with Selenium," 2011 5th Malaysian Conference in Software Engineering (MySEC), pp.217-219, 13-14 Dec. 2011.1]

[13] Automated Testing http://en.wikipedia.org/wiki/Automated_testing

[14] What is Automated Testing <http://www.softwaretestingclass.com/what-is-automation-testing/>

[15] Automation Testing versus Manual Testing <http://blogs.msdn.com/b/saraford/archive/2005/02/07/automation-testing-versus-manual-testing-guidelines.aspx?Redirected=true>

[16] Release Planning in Scrum <http://agileatlas.org/articles/item/release-planning-in-scrum>

[17] MIMOS Agile Framework - mimos-spp-proc-agile_framework

[18] PyTestFramework <https://code.google.com/p/pytestframework/>

About Author (s):

| | |
|--|------------------------------------------------------------------------------------------------------|
| | ["If you don't care about quality, you can meet any other requirement" - <i>Gerald M. Weinberg</i>] |
|--|------------------------------------------------------------------------------------------------------|

| | |
|--|----------------------------------------------------------------------------------------------------------------|
| | ["Small tests lead to code quality. Medium and large tests lead to product quality" – <i>James Whittaker</i>] |
|--|----------------------------------------------------------------------------------------------------------------|