

Automation Software Testing on Web-Based Application



Saru Dhir and Deepak Kumar

Abstract Agile testing is a software testing exercise, follows the rules of agile policy, and considers software improvement as a critical part like a client in the testing process. Automated testing is used to do this in order to minimize the amount of manpower required. In this paper, a traditional automation testing model has been discussed. A model has been proposed for automated agile testing, and an experimental work has also been represented on the testing of a Web application. Finally, outcomes are evaluated using the agile testing model, and there is a comparison between traditional and agile testing models.

Keywords Agile development and testing · Automation testing
Web application · Selenium tool

1 Introduction

A software test project can be successful, with a significant estimation and a complete execution in the software development life cycle [1]. Software testing is the main part of software development, where different level of testing is implemented, such as unit, integration, system, and acceptance testing according to the system behavior and client requirement, whereas unit and integration testing focus on individual modules, system and acceptance testing focus on overall behavior of system [2]. The adoption of the latest technology trends and development practices by the technology and software developers leads to polarization of opinions on the strengths and weaknesses of the technologies and software development practices [3]. Automation forms are one of the most important parts of agile testing; otherwise it is very difficult to keep pace with the agile development schedule.

S. Dhir (✉) · D. Kumar
Amity University, Noida 110025, Uttar Pradesh, India
e-mail: sdhir@amity.edu

D. Kumar
e-mail: dkumar8@amity.edu

The process of prioritization of defects identified by automation and their fixation during sprints is also determined at this stage. In the case of regression testing, efficiency increases with the automation testing, where test cases are implemented iteratively in the software [4].

In agile testing, generally the code is written during iteration and the testing process is done after each iteration. The developers and the testers work together to reduce the probability of error occurrences. Developers do the unit testing, and the rest system testing is done by the customer which is also known as acceptance testing and based on that clients provide their feedback which also gives a glance of excellence guarantee movement in agile [5].

2 Literature Review

In a survey result, it was revealed that only 26% of test cases are done through automation testing, which was considerably less than in comparison to the last years. The paper focused on to do more effort on automation testing and its tools [6]. According to a survey report, “State of agile development” showed that more than 80% of respondents’ organizations had adopted agile testing methodology at some level and the other half of respondents indicated that their organization is using agile testing methodology for approximate two to three years [7]. Agile software development methods are these days widely extended and established. The main reason to adopt agile testing methodology is to deliver the product within a time limit, to increase efficiency, and to easily manage the frequently changing business requirements [8]. A survey report indicated that agile testing is mostly used development process. Agile testing provides small iterations (79%), regular feedback (77%), and the scrum meeting, which were on a daily basis (71%) were the most important factors [9]. Hanssen et al. [10] said that the use of agile testing methodology is global. Evaluation and scheduling are the main concern to the accomplishment of a software growth of a project of any dimension and consequences; agile software growth of a project has been the area under discussion of much conviction.

3 Automated Testing

In Fig. 1, automation testing model is mentioned in which different steps are specified [11]. After the implementation process, the testing environment is set, plan all of the test activities. In all of the planning, different test cases are created which is reviewed by senior testers. Test cases are executed, and bug report is generated. If the bugs are not generated, iterations are completed and backlog stories are ended [12].

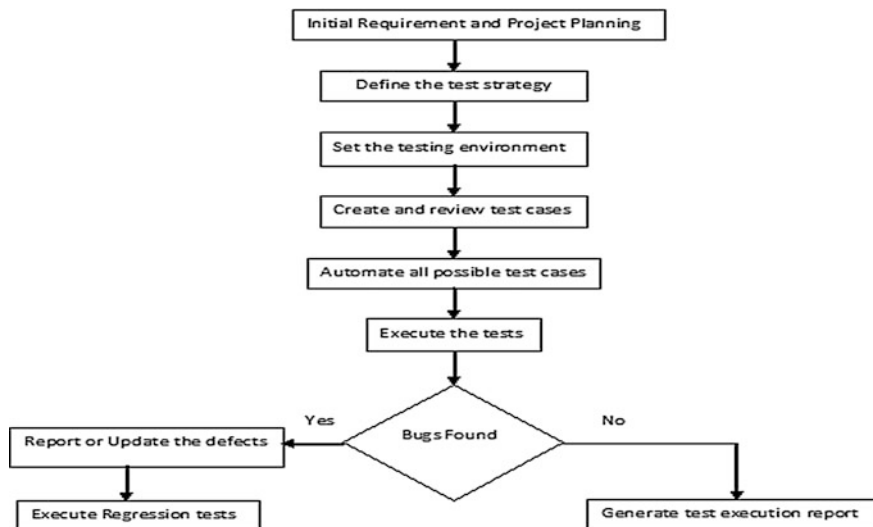


Fig. 1 Automated testing model in traditional environment

4 Proposed Method

In the agile environment, testing of the project is implemented with a proposed method to improve the accuracy of software. In an agile development, developers are also responsible for the testing. Hence, there is parallel and independent testing which is possible between developers and testers. Figure 2 is the proposed automation testing model using agile methodology.

Below algorithm is the pseudocode of automated agile testing to execute the test suites by the developers and testers. At line 2, quality assurance (QA) team initially involved in the project. Line 3 represents that the parallel testing is done by developers and testers, and stand meetings are also done to demonstrate the functionality of software. At line 4, testers generate the automation testing in different sprint cycles, and at line 5, verification test is build. At line 6 if the test suite result failed then at line 7, RCA (root cause analysis) is executed to remove the bugs. If at line 6, result successfully passed the validate the regression testing at line 10 and run the test and end. Finally at line 13 generate the report.

1. $Dd \leftarrow$ Set the delivery date
2. $QA \leftrightarrow P$ Involvement of QA team in project
3. Deploy the build in the Pre-Production environment.
4. $t \leftrightarrow d$ Testing and development both are implemented parallel.
5. $T \rightarrow$ Generate automation test suite, in the sprint cycle.
6. Release the 1st sprint and complete the BVT
7. **If** d's and t's result failed **then**

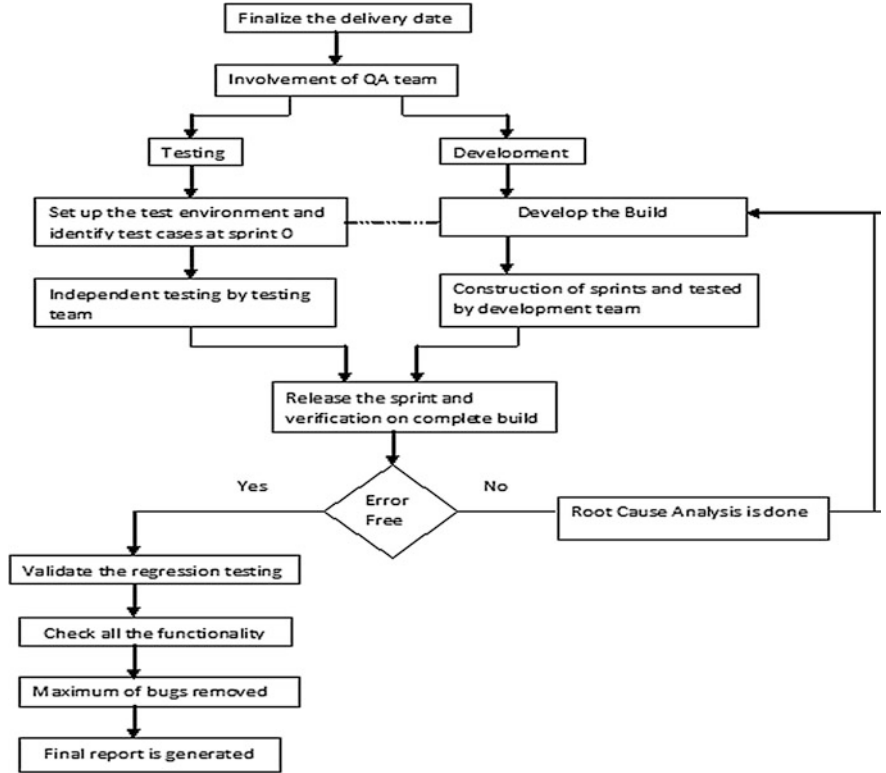


Fig. 2 Testing model using agile methodology

```
8.      do RCA ("remove the bugs in next sprint")
9.          GOTO step 3
10.     Else
11.     T → RT (validate the regression testing)
11.     Result ← T.runtest
12.     end if
13.     Generate the report
```

Whereas; *Dd*: Delivery Date; *QA*: Quality Assurance Team; *P*: Project; *T*: Tester; *D*: Development; *BVT*: Build Verification Test; *t*: testing; *d*: development; *RCA*: root cause analysis; *RT*: regression testing.

4.1 Experimental Work

During the implementation of Web application, project was developed and tested using the agile methodology. Product development and automation testing both were implemented in scrum. During the implementation in agile environment, developers and testing teams worked parallel. Automation test execution was implemented using the selenium tool in different steps. By using the steps of agile testing models, at the initial testing, bug was found as represented in Fig. 3. RCA was done, and then again a new sprint cycle was implemented. Bug was resolved as shown in Fig. 4 and validate the regression testing. Final result was evaluated as in Fig. 5, and maximum bugs were removed and a final report was generated.

Fig. 3 Failure in initial test and detect the bugs

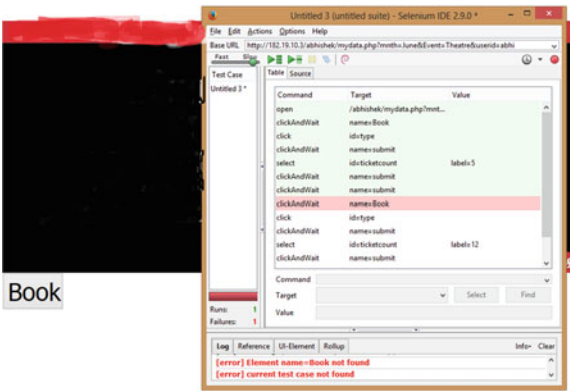


Fig. 4 Testing on a book button using selenium IDE

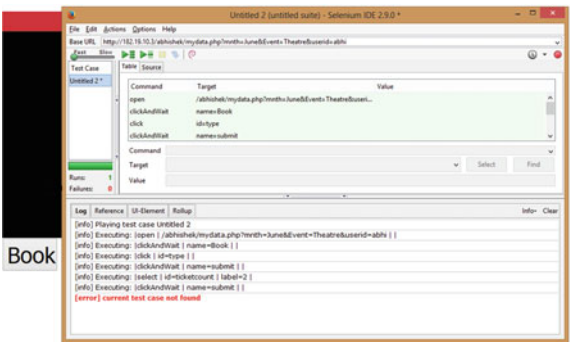
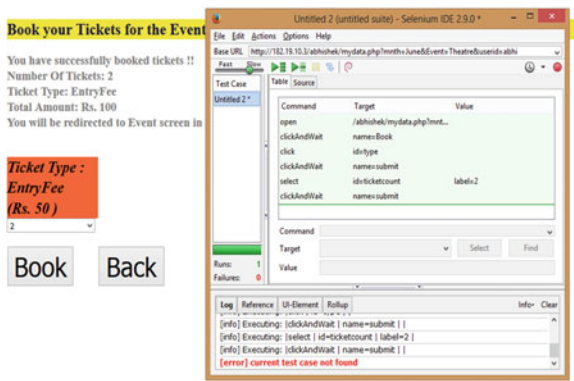


Fig. 5 Successfully booked the tickets



4.2 Comparison and Result Between Traditional and Agile Automated Testing Model

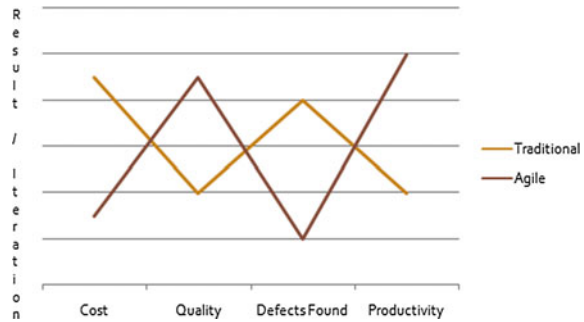
Web application was tested using the traditional and agile testing models vice versa. In the agile testing models, results were improved than the traditional model. During the testing implementation with or without agile methodology, results were observed as in Table 1:

A comparison between the testing in agile and traditional environments is represented in Fig. 6. Results in figure are based on different parameters per iteration. Parameters are cost, quality, detected bugs, and productivity.

Table 1 Comparison between agile model and automated testing model

Agile testing model	Automated testing model
According to the delivery date of the project, testing was done in parallel to the development process	Testing was implemented after the completion of development process
Due to the parallel testing from the initial stage of the development, less number of defects were evaluated at the end	Testing was done without the customer’s involvement and took more time for customer satisfaction
As regression testing was executed frequently in a single sprint, it saved time and cost	After the completion of development process, test cases were created followed by automated test scripts
Productivity and quality of product were improved with the sequent involvement of customers	Less productivity with the low quality of product

Fig. 6 Testing parameters using agile implementation and traditional implementation



5 Conclusion

Paper concludes that the automation testing not only increases the test coverage but also reduces the cost and improves the delivery of the product. In this paper, an experimental work was discussed in which an automation testing was implemented on a Web application using agile environment and traditional development. The proposed agile testing model worked with productization team in a planned and organized manner to deliver the products in sprint. At the end, there is comparison between agile and traditional automated testing models, which specifies the results through the agile testing is better than traditional testing.

References

1. Sharma, M.A., Kushwaha, D.S.: A metric suite for early estimation of software testing effort using requirement engineering document and its validation. In: Computer and Communication Technology (ICCCT) 2nd International Conference, pp. 373–378 (2012)
2. Spillner, A., Linz, T., Schaefer, H.: Software Testing Foundation: A Study Guide for the Certified Tester Exam, 3rd ed. Rocky Nook (2011)
3. Aggarwal, S., Dhir, S.: Ground axioms to achieve movables: methodology. *Int. J. Comput. Appl.* **69**(14) (2013)
4. Karhu, K., Repo, T., Taipale, O., Smolander, K.: Empirical observations on software testing automation, international conference on software testing verification and validation (ICST'09). IEEE Computer Society, Washington, DC, USA, pp. 201–209. DOI = 10.1109/ICST.2009.16 (2009)
5. Aggarwal, S., Dhir, S.: Swift tack: a new development approach. In: International Conference on Issues and Challenges in Intelligent Computing Techniques, IEEE (2014)
6. Kasurinen, J., Taipale, O., Smolander, K.: Software test automation in practice: empirical observations. *Adv. Softw. Eng.*, Article 4 (2010)
7. http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf
8. Rai, P., Dhir, S.: Impact of different methodologies in software development process. *Int. J. Comput. Sci. Inf. Technol.* **5**(2), 1112–1116 (2014). ISSN: 0975-9646
9. West, D., Grant, T.: Agile development: mainstream adoption has changed agility. Forrester Research Inc. (2010)

10. Hanssen, G.K., Šmite, D., Moe, N.B.: Signs of agile trends in global software engineering research: a tertiary study. In: 6th International Conference on Global Software Engineering, pp. 17–23 (2011)
11. Collins, E., Macedo, G., Maia, N., Neto, A. D.: An industrial experience on the application of distributed testing in an agile software development environment. In: Seventh International Conference on Global Software Engineering, IEEE, pp. 190–194 (2012)
12. Mattsson, A., Lundell, B., Lings, B., Fitzgerald, B.: Linking modeldriven development and software architecture: A case study. *IEEE Trans. Softw. Eng.* **35**(1), 83–93 (2009)