International Journal of Future Generation Communication and Networking
Vol. 13, No.2s, (2020), pp. 1694–1699

Test Case Management Tool

[1]Dhruvi Modi, [1]Alisha Bhale, [1]Karan Borana, [1]Vaishnavi Chatap,

[1]*BE-Computer, VIIT Pune*
[2]*L.A.Deshpande, Assistant Professor, Computer Engineering, VIIT Pune*

Abstract

Software Testing is one of the crucial activities in developing quality software. Although a range of software testing techniques has been developed to efficiently identify bugs in source code, these techniques are not always fully employed in practice. This has many explanations, including the challenge of understanding the complexity of handling all the test cases for large-scale projects. Test case management involves organizing the testing process in a systematic manner. To be successful, test case management requires a high degree of discipline to accommodate the large volume of elements under consideration. This paper aims at the development of a powerful test case management tool. Our proposed methodology will manage the test cases efficiently thereby reducing the effort required by the testing team, saving cost and approximating the complexity of them. Tracking the status of every test case, from its engineering to complexity estimation, will be done efficiently. The main feature of the system is that it will be estimating the complexity level of the test cases based on a number of steps, verification points and data entry points in the test case.

Keywords *- Software Testing, Web-based Software Tool, Complexity Estimation, Status Tracking, Verification points, Data entry points*

I. INTRODUCTION

People involved in a software project want to know when the testing is done and the quality of the testing. It is critical that the test execution is effectively monitored in order to be able to answer that question. So, this is done by collecting test data, or measurements, showing the test progress. The metrics aid in determining when corrections are required to ensure progress. Further, the testing team can estimate a release date for the program using these metrics [5].

To improve test performance and enhance test repeatability, commercial vendors have developed various testing tools. These software test management tools have dominated the market over the past decade and have been widely adopted. Yet increasingly, IT organizations are acknowledging Open Source test management tools as reliable and implementing them rapidly. Open source test management tools reduce cost, increase tester efficiency, test in-house developed applications and so on [3].

This paper discusses the development of a web-based test management tool that will be used by testers, test leads & managers of an organization. It will follow a centralized test management concept that will aid in easy communication and easily allow managing test cases and test results as well as maintain a standardized test case format. The complexity of the underlying database is entirely hidden from the system end-users. Also, the test scenarios will be managed at different user levels like a manager, tester and test lead. An Individual's performance will be reflected, through the dashboards, to the manager and project leader. This tool will make the tracking of the status of individual tests and projects easier with the help of the dashboard. The cost incurred for the management of the testing process will also be reduced as this tool will be open source. The major advantage of this web-application will be its ability to estimate the complexity of any test case as well as the time required for the execution of a test case. This estimation will help the team in adjusting the release date so as to ensure timely delivery to the client.

ISSN: 2233-7857 IJFGCN

The rest of the paper is structured as follows: Section II explains the related work done in this area, Section III describes the proposed methodology and Section IV shows the results followed by the conclusion.

II. RELATED WORK

From the moment the development of software begins, it's testing also starts. As the development progresses the complexity in testing increases. While spreadsheets can be used as a way for a small team to handle test cases, as the team expands, it can quickly turn into a huge burden. It becomes difficult to keep track of various test scenarios that are being handled by different testers. This can lead to miscommunication and ultimately resulting in client dissatisfaction [9].

Manually managing the testing process has various limitations :

- Traceability is not evident between specifications and the test cases.
- Absence of a shared database for storing the results.
- Difficulty in managing the project activities among the team members.
- Tracking individual and team performance is hard [1].

In order to overcome the limitations of manual testing management, several tools have been developed over the past decade. These tools satisfy the basic functionality of managing the test cases as well as provide some additional features that aid the testing team.

TABLE I

Some existing test case management tools [6-8]

Product Name	Key Supremacy	Key Weakness
TestRail	Clear metrics, and analyses in real-time	Subscription cost increases as the number of users increases
TestLodge	Easy to use and offers templates for the test cases	Can be inefficient if applied to large teams
Tricentis qTest	Comprehensive test case management features	Not very affordable for small teams

III. PROPOSED METHODOLOGY

The proposed methodology reflects the management of the test cases throughout the whole testing process. The methodologies basically include the two major aspects, the estimation of the complexity of test cases and automation.

The process will start with the requirement gathering from the client which basically includes the test case scenario, the number of steps to be covered to reflect the required result. There are three main user roles that will hold the core functionality to be delivered. The proposed methodology aims at delivering the load balancing facility for as well as the performance tracking for an individual user role. The overall methodology delivers core functionality for the smooth processing of the testing phase.

ISSN: 2233-7857 IJFGCN

The core functionality of this methodology lies in the estimation of the complexity of each test case. There are three main attributes to calculate the complexity level. The result will reflect on three different statuses.

The parameters are as follows:

1. The number of steps.
2. The data entry points.
3. The verification points.

TABLE II
Estimation of test case complexity

TC Complexity	Steps in TC	Verification Point	Data Entry Point	Per Day Estimate
Simple	<=10	<=8	<=10	3
Medium	<=15	<=12	<=15	2
Complex	30+	20+	20+	1

A. Technologies

Front-end technologies used are HTML, CSS (Cascading Style Sheet), Bootstrap and JavaScript for User-Interface development.

Back-end technologies used are Node.js for server-side scripting and MySQL for Database.

B. Architecture

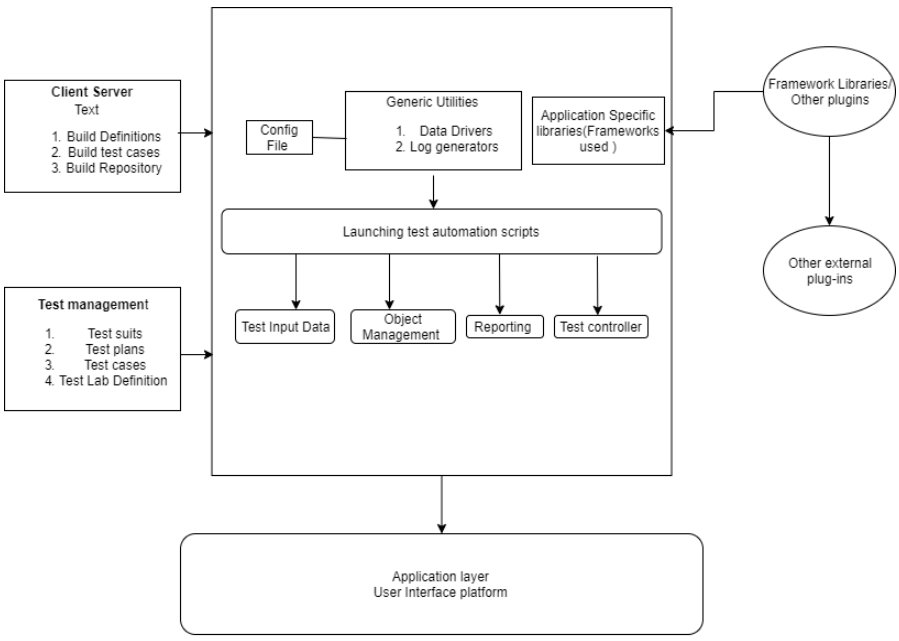

Figure 1. The architecture of the system

Software Architecture Design above describes the basic software structure by separating functional areas into layers. It shows how this software might interact with its user, external system, data sources, and services.

ISSN: 2233-7857 IJFGCN

Client-Server Texts are the input that a system will get from a client-side in the form of Build Definition, Build test cases and Build Repository. The software will further manage the Test input by creating Test suites, Test plans, Test cases, and Test Lab Definition.

The config file is used to manage various settings that define a web system. It uses various Data driver to manage data in a system and help operating systems and devices to communicate data between them. Log generators are used to maintain a log of individual users at the backend[10 11]. Various Frameworks which are software libraries that provide a fundamental structure to support the development of the application for a specific environment are used. Further launching of these Test Automation Scripts will provide services for Test Input Data, Object Management, Reporting, and Test Controller. These services are then used by individual users with the help of the User Interface platform provided by the Application layer which provides with dashboard, input fields, datasheet, etc. to the user.

IV. RESULTS

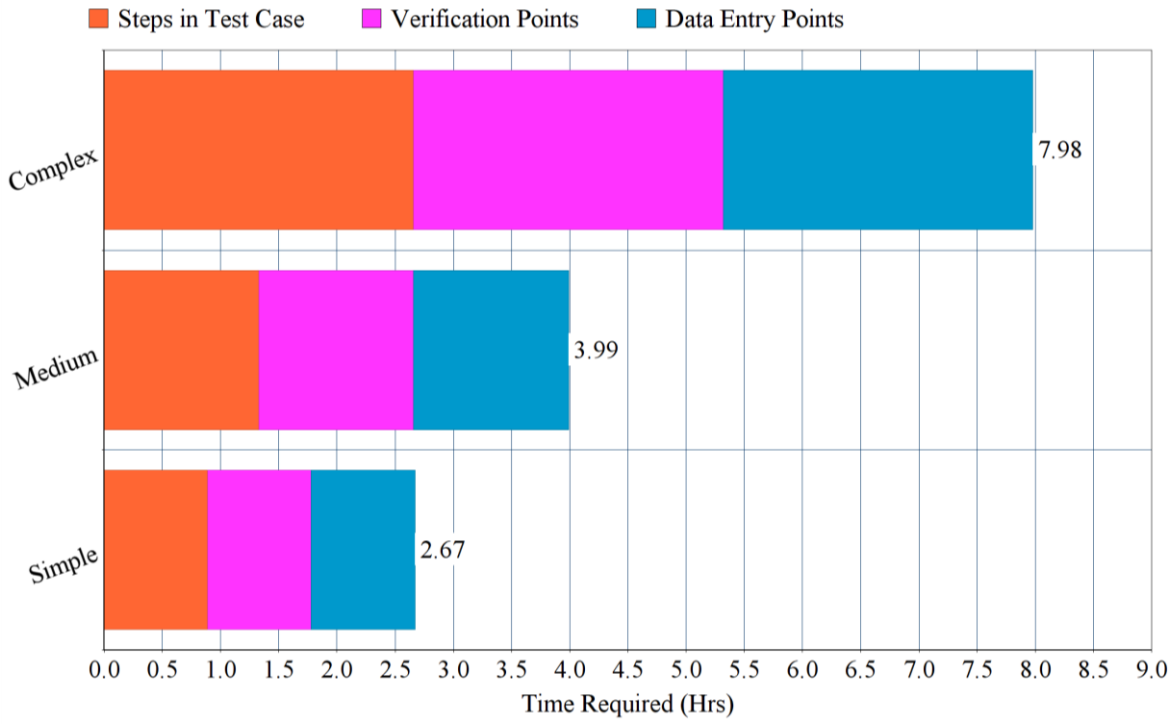

Figure 2. Time vs. Complexity (For a single Test Case)

Conclusion of the above graph - The above graph signifies the approx time taken by three different complexity of Test Cases. If a user wants to predict the approximate time required by a particular Test Case then the above graph can be very handy in predicting the time.

According to TABLE II of Test Case Complexity given above, 3 Simple Test Case can be performed in a day. So considering an 8-hour workday, each test case gets approximately 2.66 hours. Therefore dividing the 2.66 hours for all the 3 sections of a test case, i.e Steps, Verifications pts, and Data entry pts take 0.89 hrs each approximately. Similarly, all the sections take 1.33 hrs and 2.66 hrs for medium complexity and a hard complexity test case respectively.
Example - So if a Test Case which contains Steps = 12, Verification Points = 10, Data Entry Points = 7 , then

Time Required is -

ISSN: 2233-7857 IJFGCN

Steps = 1.33 hrs (as Steps greater than 10)
Verifications Pts = 1.33hrs (as Verifications Pts greater than 8)
Data Entry Pts = 0.89 hrs (as Data Entry Pts less than 10)
Total Time = 1.33 + 1.33 + 0.89 = 3.55 hrs approx.

So the above example Test Case can take 3.55 hrs approximately.

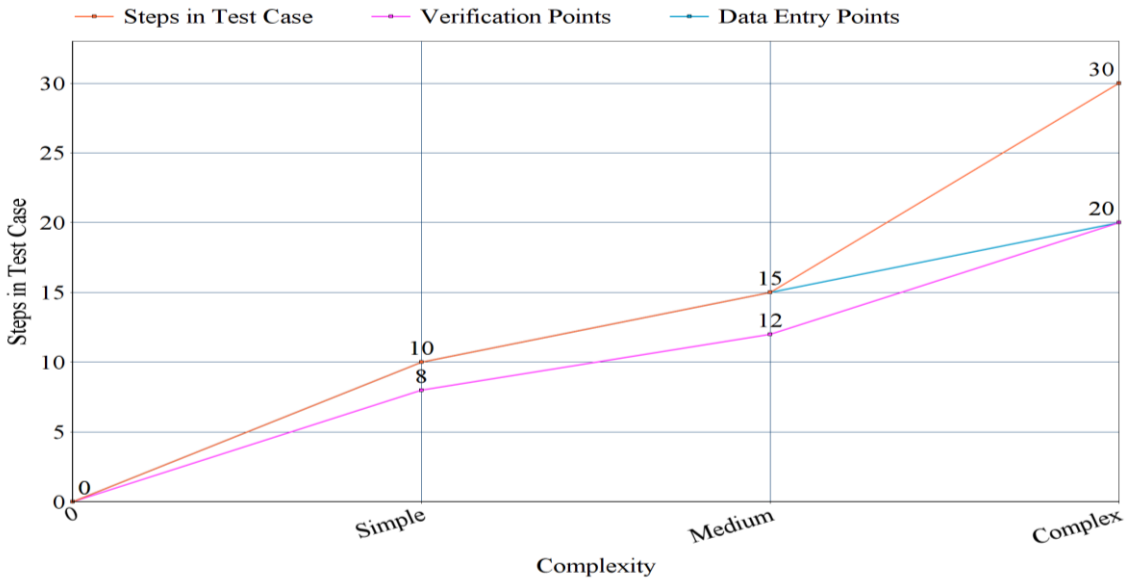

Figure 3. Complexity vs. Steps in Test Case

The above graph significantly describes how the three parameters i.e. Steps of Test Case, Verifications Points, Data Entry Points affect the complexity of a particular Test Case. By the above graph data, one can easily get a brief idea about the complexity of a particular Test Case by observing the variations in the values.

V. CONCLUSION

This paper discusses a web-based software tool that will be used for the management of the test cases. This tool will allow the testing team to efficiently manage the test cases, reduce the cost of testing and avoid chaos. The important feature of estimating the complexity of the test cases will provide an approximate test case completion time. With the help of this estimation, changes can be made to the deadline given to the client so as to ensure timely delivery. Further features can be added and more aspects of the testing phase can be automated. A lot of research is ongoing concerning this domain.

REFERENCES

[1] Sheena Kukreja, Abhishek Singhal, Abhay Bansal, "A critical survey on test management in IT projects", International Conference on Computing, Communication and Automation (ICCCA2015)
[2] Ahmed Ibrahim Safana; Suhaimi Ibrahim, "Implementing Software Test Management Using SpiraTeam Tool", Fifth International Conference on Software Engineering Advances, Aug 2010
[3] K. Saravanan, E. Poorna Chandra Prasad, " Open Source Software Test Automation Tools: A Competitive Necessity ", International Journal of Management & Development, Vol.03, Issue 06 (2016) pg103-110
[4] Tauhida Parveen, Scott Tilley, George Gonzalez, "A Case Study in Test Management"
[5] J.B.A Gemunu Priyadarshana, "Software Test Management Tool", A dissertation submitted for the Degree of Master of Information Technology, University of Colombo School of Computing 2017
[6] "TestRail tool", https://www.gurock.com/testrail
[7] "TestLodge tool", https://www.testlodge.com/
[8] "Tricentis qTest", https://www.tricentis.com/products/agile-dev-testing-qtest/

ISSN: 2233-7857 IJFGCN

[9] “Understanding test case management”, https://www.getzephyr.com/insights/understanding-test-case-management#
[10]Dhumane, A., & Prasad, R. (2015). Routing challenges in internet of things. CSI Communications.
[11] Dhumane, A. V., Prasad, R. S., & Prasad, J. R. (2017). An optimal routing algorithm for internet of things enabling technologies. International Journal of Rough Sets and Data Analysis, 4(3), 1–16.

ISSN: 2233-7857 IJFGCN