Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey on Automated Software Testing Tools

Nithya TM
HOD, Dept. of CSE
K. Ramakrishnan College of Engineering, Trichy, TN, India

Amudha L
Asst. Prof, Dept. of CSE
K. Ramakrishnan College of Engineering, Trichy, TN, India

Ramya J
Asst. Prof, Dept. of CSE
K. Ramakrishnan College of Engineering, Trichy, TN, India

Abstract: ***In the present day scenario there is need for accelerated software development. Software testing is a technique aimed at evaluating an attribute or capability/usability of a program or product/system and determining that it meets its quality. Although crucial to software quality and widely deployed by programmer & testers, software testing still remains an art, due to limited understanding of the principles of software. Identifying the types of testing that can be applied for checking a particular quality attribute. All types of testing cannot be applied in all phases of software development life cycle. Which testing types are applicable in which phases of life cycle of software development is also summarized.***

Keywords: SDLC, Testing, Quality, Efficiency, Software.

I. INTRODUCTION

Software testing is an integral part of software development process. Software testing is analyzing a system or a component by providing defined inputs and comparing them with the desired outputs to check the discrepancies between the desired and actual outputs and correct them. Basically software testing can be divided into two categories. They are Manual testing and Automated software testing.

By integrating automated testing into the software development program, you gain a number of benefits:

- Reduced cycle time by decreasing product and integration test time.
- Improved quality
- Standardized testing and reproducible results.

Manual software testing is as the name suggests done manually that is it requires human input, analysis and evaluation. Automated software testing is the automated version of manual software testing.

Software testing is a set of activities to find out the Errors and to provide the quality product or software.It also verifies and validate the program whether program working correctly without any errorsSoftware development is the process of coding functionality to meet defined end-user needs. While Software testing tends to be considered a part of development, it is really its own discipline and should be tracked as its own project. Software testing, while working very closely with development, should be independent enough to be able to hold-up or slow product delivery if quality objectives are not met. The objective of software testing is to find problems and fix them to improve quality.

The main purpose of testing can be quality assurance, reliability estimation, validation or verification. The other objectives or software testing includes.

- Testing is a process to identify the correctness and completeness of the software.
- It can be tested if it works the more efficiently .
- The software can be controlled more the testing can be automated and optimized.
- The fewer the changes, the fewer the disruption to testing.
- A successful test is the one that uncovers an undiscovered error.

II. ANALYSIS

Software Quality

Structural quality is evaluated through the analysis of the software inner structure, its source code, at the unit level, the technology level and the system level, which is in effect how its architecture adheres to sound principles of software architecture . In contrast, functional quality is typically enforced and measured through software testing. Quality cannot be achieved by assessing an already completed product. The aim therefore, is to prevent quality defects or deficiencies in the first place, and to make the products assessable by quality assurance measures. Some quality assurance measures include: structuring the development process with a software development standard and supporting the development process with methods, techniques, and tool.Testing is done in many phases depending upon the requirements of the software being developed. Testing can be both manual and automated depending on what suits the requirements. Testing is a planned process with care taken especially on what test has to be done when.

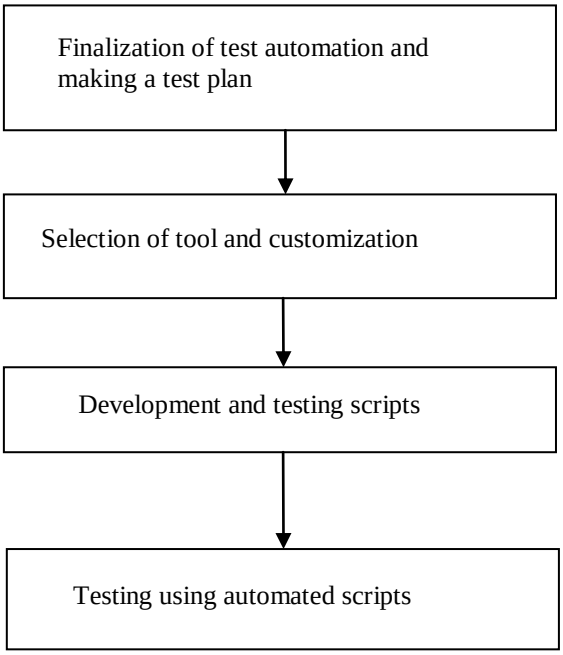
© 2016, IJARCSSE All Rights Reserved

2.1 Requirements for Automated Software Testing:

All the tests are not automated. There are certain requirements for a test to be automated. They may be financial restrictions or limited man power and many. There a few basic questions whose answers could give you an idea whether the test has to be automated or not. They are as follows

1. Can the test sequence of actions be defined?
2. Is it necessary to repeat the sequence of actions many times?
3. Is it possible to automate the sequence of actions?
4. Is the behavior of the software under test the same with automation as without?
5. Do you need to run the same tests on multiple hardware configurations?

2.2 Test Automation Process Life Cycle Diagram



2.3 Finalization of Test Automation and Making a Test Plan:

During this phase it is decided which tests can be automated. The test plan is made. The test automation process is similar to the software development process. It takes similar effort to automate a test. It follows the same cycle as in development of a software product. The plan is made by taking into consideration amount of time required and number of people required and who does it.

III. SELECTION OF TOOL

Test tool selection is a very important part of test automation. This requires the study of the scope of testing and the test plan .It is also needed to know whether the test tool meets the test suite requirements for the particular product and version. The important factors that also come into picture are reusability, reliability and cost. This is done to see if they can get the maximum benefit out the product being made, bought or customized

The test tool should support

1. Scripting interface
2. Facility to give valid and invalid input
3. Result comparison
4. To give the verdict

The test manager has to look for tools which are already available and check which one is suitable. If the manager does not find any tool that is not suitable than he think of taking an available tool and customizing it according to the needs. If this not possible then only should the manager decide to develop a new tool.

3.1 Development and Testing pf Scripts:

Development process includes both development and testing of scripts. This mainly depend on two factors

1. The skill of the person who is writing the scripts
2. On the flexibility of the test tool for developing for all valid and invalid scenarios

It is always advisable to develop scripts in a modular way .This modular approach towards writing scripts helps in reusability of the module in the different scripts. This approach will use less time and will be enable the use of the scripts across different releases. The scripts should always be tested before being put to use .They can be tested on an already released version of the software. This is done to avoid problems which may arise during the testing process.

© 2016, IJARCSSE All Rights Reserved

3.2 Testing Using Automated Testing Scripts

This is the face where the actual testing is done. It solely depends on the test plan when a test is automated which one are automated and in which part of the development cycle these tests are done. The scripts that have been made are developed in such manner that if there is another version release of the software product then these scripts could be used for testing purpose for that version too. These thinks have to be kept in mind by the test manager when he plane for test automation.

IV. TOOLS FOR AUTOMATED SOFTWARE TESTING

For many test managers, the decision of which testing tools to use can cause confusion. The first decision to make is which category of tool to use—one that tests specific units of code before the application is fully combined, one that tests how well the code is working as envisioned, or one that tests how well the application performs under stress. And once that decision is made, the team must wade through a variety of choices in each category to determine which tool best meets its needs. Evaluating your needs will narrow down your short list very rapidly. There are many options in market to choose.

Depending on what sort of test it will be used for the choice may become easy. The following general categories categorize the tests and tools which makes it easy to choose

1. Developer oriented tools
2. Functional testing tools
3. Load testing tools
4. Performance monitoring and maintenance tools

4.1 Developer Oriented Tools:

Tools in this category, also called component testing tools or unit testing tools, test individual software components or groups of related components, helping isolate and rectify problems as early in the development process as possible. Areas of testing include memory analysis, function analysis . The important tools in this area are

- DevPartner Studio from Compuware Corp.
- PurifyPlus from IBM Rational.
- Optimizeit Enterprise Suite from Borland Software Corp.
- C++Test,Jtest and TEST from Parasoft.

4.2 Functional Testing Tool:

Tools in this class help verify that applications will work as expected. It allows developers to record an existing application and modify scripts to meet changes in an upcoming release. They also provide for regression testing on the new release using the test scripts developers have captured up until that point. The important tools in this area are

- WinRunner from Mercury interactive Corp.
- Astra QuickTest Mercury interactive Corp.
- SilkTest from Segue Software Inc.
- RationalSuite TestStudio from IBM Rational.

4.3 Load Testing Tools:

This category, also called performance or stress testing tools, tests what happens to the code as the application scales with multiple users in a simulated environment. The tools test, among other things, whether performance degrades as the load is increased. Once bottlenecks are found, these tools can determine the source of the problem and begin fixing it. The important tools in this area are

- LoadRunner from MercuryInteractive.
- SilkPerformer from Segue Software Inc.
- Testperspective and Load test editor from Keynote Systems Inc.
- QACentre from Compuware.

4.4 Performance Monitoring and Maintenance Tools:

All the tools in this category are used after the application is already in production by examining how the application is working in a real-world environment. In essence, these tools monitor the production environment to ensure that all requirements and defined thresholds are continually being met.

The important tools in this area are

- Vantage from Compuware.
- VTuneAnalyzer from Intel Corp.
- Optimizeit Enterprise Suite from Borland Software Corp.
- OneSight from Empirix

4.5 Practical Features of Automated Software Testing:

1. Run all day and night in unattended mode
2. System continues running even if a test case fails

© 2016, IJARCSSE All Rights Reserved

3. Keep the automated system up and running at all costs
4. Recognize the difference between hard and soft errors
5. Write out meaningful logs
6. One point maintenance
7. Easy to update reusable modules
8. Text strings stored in variables easy to find and update
9. Written in an English-like language easy to understand
10. Automated most important business functions first.
11. Quickly add scripts and modules to the system for new features
12. Don't waste time with very complex features, keep it simple
13. Collect other useful information such as operating system and CASE tool message
14. Track components of the automated testing system in a database
15. Track reusable modules to prevent redundancy
16. Carefully test the testing system
17. Keep track of tests coverage provide by automated test suites
18. Track which test cases are automated and which are manual

V. CONCLUSION

Software testing still remains an art, due to limited understanding of the principles of software. Identifying the types of testing that can be applied for checking a particular quality attribute is the aim of this thesis report.Automation of software testing is not an easy process. It involves a lot effort. Automation is also a expensive process in terms of direct finance and the people required for. If the process of automation is done in very definitive manner and with all aspects considered it can be asset to the company.Automation for initial stages will be difficult but at later stages,it pose a vital role in an organization.

REFERENCES

[1] http://www.origsoft.com/Issues3-buscase-4save.htm
[2] http://www.sqa-test.com/ATS_Tips.html
[3] http://www.sqa-test.com/articles.html
[4] http://www.origsoft.com/Issues2-comptest-1cov.htm
[5] http://www.cssttechnologies.com/Test%20Requirements%20for%20automated%20Testing.htm
[6] http://www.io.com/~wazmo/succpap.htm
[7] http://www.hssworld.com/whitepapers/whitepaper_pdf/test_automation.pdf
[8] http://www.stsc.hill.af.mil/crosstalk/2002/05/rice.pdf
[9] http://www.testing.com/writings/automate.pdf
[10] http://www.qtp
[11] http://www.tutorialspoint.com/qtp/
[12] http://www.softwaretestinghelp.org

© 2016, IJARCSSE All Rights Reserved