Software Engineering Automated Software Testing

Ramya. S
Department of Master of Computer Applications,
Global Institute of Management Sciences
Bangalore, India

Abstract:- **Releasing software in frequent and shorter release cycles necessitates a testing approach that helps create automated tests for multiple devices, platforms, and environments easily and quickly. After the manual test cases have been created decide with your manager which test cases should be automated. Use the Automation Checklist to assist you in deciding what tests to automate. If you have enough manpower you may want to have a test plan team and an automation team. The test plan team would develop tests manually and the automation team would decide which of the manual tests should be run. The automation team would be responsible for assuring that the test can be successfully and cost effectively automated. Different automated testing tools are listed out to fit different applications needs using which the software can be tested in a faster pace.**

I. INTRODUCTION

First I would like to quote a saying about automated testing:

"Success in test automation requires careful planning and design work, and it's not a universal solution. ... Automated testing should not be considered a replacement for hand testing, but rather as an enhancement." (Software Testing with Visual Test 4.0, forward by James Bach, pg. vii)

Automation activity during the design phase. As soon as the functional specification is written, create all test cases so that they can be run manually. Yes, that's right, manually! These manual tests are step-by-step "pseudo" code that would allow anyone to run the test. The benefits of this approach are:Your test cases can be created BEFORE ever seeing the software's user interface (UI).

II. EASE OF USE

Sometime during the design phase, as soon as the design is firm enough, you'll select the automation tools that you will need. You don't have to decide exactly which tests need to be automated yet, but you should have an idea of the kinds of tests that will be performed and the necessary capabilities of the tools. That determination is easier as the software gets closer to the code complete phase. Your budget and available resources will begin to come into play here.

III. DESCRIPTION

Automation activity during the Code Complete Phase

Although the UI may still change, QA can begin writing Automatic test cases. The tests that should be written at this point are breadth tests that tell the status of the overall software product. Don't write tests which stress the product until you get close to Alpha. The product will probably break very easily. Some acceptance (or "smoke") tests should also be created to give a quick evaluation of the status of a particular build. Before reaching the Alpha phase there should also be tests written to test the Installer, boundary (or stress tests), compatibility (hardware and OS), performance, and interoperability.

Automation during alpha phase:

So At this point you have done the tasks that need to be done in order to reach Alpha. That is, you have all your compatibility, interoperability, and performance tests completed and automated as far as possible. During Alpha you'll be running breadth tests every build. Also you'll run the compatibility, interoperability, and performance tests at least once before reaching the next milestone (beta). After the breadth tests are run each build, you'll want to do ad hoc testing as much as possible. As above, every bug should be associated with a test case to reproduce the problem.

Automation during beta phase:

What is the Beta Phase? The product is considered "mostly" bug free at this point. This means that all major bugs have been found. There should only be a few non-essential bugs left to fix. There's no more time left to develop new tests. You'll run all of your acceptance tests as quickly as possible and spend the remaining time on ad hoc testing. You'll also run compatibility, performance, interoperability and installer tests once during the beta phase.

Remember that as you do ad hoc testing every bug should have an associated test case. As bugs are found during ad hoc testing, new test cases should be created so that they can be reproduced easily and so that regression tests can be performed when we get to the Zero Bug Build phase.

Automation Activity during the Zero Defect Build Phase

Run regression tests. Regression testing means running through your fixed defects again and verify that they are still fixed. Planning for regression testing early will save a lot of time during this phase and the Green Master phase.

What is the Green Master Phase? Green Master is sometimes referred to as the Golden Master or the final candidate. The product goes through a final checkout before it is shipped (sent to manufacturing).

Automation activity during the Green Master Phase

After running general acceptance tests, run regression tests. You should run through your fixed defects once again

to verify that they are still fixed. Planning for regression testing early will save a lot of time during this phase.
Here I will list out few automated software available in market which are used for testing :-

IV. TYPES OF AUTOMATED TOOLS

AUTOMATED WEB TESTING TOOLS

Automatically testing your web application is a good way to ensure that new versions of your application don't introduce bugs and regressions. Automation of your web application testing also allows your development team to make changes and refactor code with more confident, as they can quickly verify the functionality of the application after every change.
Open Source

Selenium

Selenium is a popular automated web testing tool and helps you automate web browsers across different platforms. Selenium has the support of some of the largest browser vendors who have taken steps to make Selenium a native part of their browser.

Open Source
Watir
Watir is a set of Ruby libraries for automating web browsers and allows you to write tests that are easy to read and maintain. Watir drives browsers the same way people do (it clicks links, fills in forms, presses buttons etc.) and also checks results such as whether expected text appears on the page.

Open Source
Windmill
Windmill is a web testing tool designed to help testers automate and debug web applications. It comes with a cross-browser test recorder, JavaScript integration and an interactive shell to automate web browsers.

Commercial
Ranorex
Ranorex allows you to automate your web application testing (among other things) and both record user interactions and play them back to execute your tests. Ranorex is one of the more popular commercial tools to build and run automated web and GUI tests.

Open Source
SoapUI
SoapUI is a cross-platform functional testing tool. It has been specifically designed to help automatically test APIs such as SOAP and REST interfaces to ensure the interoperability of different applications.

Open Source
Sahi
Sahi is a tool for automation of web application testing. Sahi is available as a free open source edition as well as a commercial Pro edition.

Commercial
Tellurium
Tellurium is a web automation tool that allows you to design and write your automated tests using plain English without any scripting or programming experience.

AUTOMATED GUI TESTING TOOLS

Building robust automated GUI tests for desktop applications (e.g. on Windows or Mac systems) is quite difficult, as small changes to the user interface can often result in broken tests. The following tools help you build and execute robust GUI tests for various platforms and operating systems.

Commercial
Squish
Squish is a GUI testing tool for various platforms, including QT, native Windows and Mac applications. Squish allows testers and developers to build automated tests using familiar scripting languages such as JavaScript, Perl, Python and Ruby.

Commercial
Ranorex

Ranorex allows you to automate your desktop applications (among other things) and both record user interactions and play them back to execute your tests. Ranorex is one of the more popular commercial tools to build and run automated GUI and web tests.
Commercial
TestComplete

TestComplete is an automated testing tool for the Windows platform. It allows you to record, script and run GUI tests for applications built using different frameworks and languages, such as .NET or C++.
Commercial
Test Studio
Test Studio is an automated functional and load testing tool that helps you test applications on various platforms built using different frameworks and tools.
software applications and enterprise teams. It can be used to automate different application types, such as .NET, Java and Flash applications.
Commercial
eggPlant :eggPlant is a GUI test automation tool for professional

V. ADVANTAGES OF AUTOMATED SOFTWARE TESTING:

1. Automated Software Testing Saves Time and Money
2. Vastly Increases Your Test Coverage
3. Testing Improves Accuracy
4. Automated QA Testing Helps Developers and Testers
5. Automation Does What Manual Testing Cannot
6. QA and Dev Team Morale Improves

VI. CHALLENGES OR DRAWBACKS OF AUTOMATED SOFTWARE TESTING

1. Proficiency is required to write the automation test scripts.
2. Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to deadly consequences.
3. Test maintenance is costly in case of playback methods. Even though a minor change occurs in the GUI, the test script has to be rerecorded or replaced by a new test script.
4. Maintenance of test data files is difficult, if the test script tests more screens.

VII. FUTURE ENHANCEMENT

The world of software testing and quality control is moving towards a paradigm shift. As the amount of data increases and the need for more efficiency presses itself, the focus is being set more on test automation. As established worldwide, the process of software testing is a never ending, always improving process. There is no such thing as a totally bug-free, perfect application. However, a well-tested application is always possible. And it is this scope for improvement that keeps the spark in testing alive.

VIII. CONCLUSION

Successful tool automation depends not only on tools but also on a standard testing process and the right test team roles, duties and skills. Tools, process and test team are the three essential legs of the test automation stool. Moreover, the automation test team needs to have a blend of testing, programming, and tool knowledge. If an organization wants to reap the automation benefits promised by tool vendors, it needs to use the tool as a complement to manual testing. It also means adopting a strong test methodology and training the test team on the ins and outs of the selected tool in an organization's unique development environment.

REFERENCES

[1] Testing Computer Software (By: C. Kaner) Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing (By: Rex Black)
[2] Software Engineering, A practitioner's Approach by Roger S. Pressman, McGrawHill International Edition, 6th Edition
[3] Software Engineering by Sommerville, Pearson Education, 7th edition
[4] Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality (By: Elfriede Dustin)
[5] https://smartbear.com/learn/automated-testing/ *info.dogpile.com/ alhea.com/**List**+Of+**Automation**+**Test**+**Tools*** www.load**testingtool**.com/
[6] https://ihe.net/**Testing_Tools**/
[7] www.**software**testinghelp.com
[8] www.testingstuff.com/**references**.html
[9] www.computersciencezone.org/**software**-quality-assurance/
[10] www.**testing**excellence.com/types-of-**software**-**testing**-complete-**list**/