

Regression Testing Analysis Using Automated CASE Tools

¹Dr. Rajender Bathla , ²Dr. Gundeep Tanwar

¹HCTM, Kaithal , ²BRCMCET, Bahal

¹dr.bathla@gmail.com , ²tgundeep@brcm.edu.in

Abstract: The main objective of this paper focuses on the importance of automated software testing associate with software testing techniques in software engineering. In which we consider, categorized and enlighten on the software testing using current scenario of testing automation. The solution of this problem leads to the new approach of software development known as software testing in the Information Technology world. Software test automation is the process of automating, the steps of manual test cases using an automated tool or utility to shorten the testing life cycle with respect to time. Regression testing is commonly used to efficiently test the system by systematically selecting the appropriate minimum suite of tests needed to adequately cover the affected change. Common methods of regression testing include rerunning previously run tests and checking whether previously fixed faults have re-emerged.

Keywords: Regression Testing, Software Testing, Manual Testing, Automated Testing Tools (CASE)

I INTRODUCTION

Software automated testing is the process of executing a program with the intention of finding errors in the code. It is the process of exercising or evaluating a system or system component by manual automatic means to verify that it satisfies specified requirements or to identify differences between expected and actual results [3],[4] Software Testing should not be a distinct phase in System development but should be applicable throughout the design development and maintenance phases. ‘Software testing is often used in association with terms verification & validation’ [5]. ‘Software testing is the process of executing software in a controlled manner, in order to answer the question: Does the software behave as specified. One way to ensure system’s responsibility is to extensively test the system. Since software is a system component it requires a testing process also.

II WHY MORE AUTOMATION (CASE)

The use of automatic tools in software testing is too often limited, in current industrial practice, to the running of the tests, sometimes including a measurement of the coverage, and the comparison of the results with expected results. It is up to the user, sometimes aided by a test script editing tool, to design the test cases and calculate the test data (inputs and expected outputs) for each case. As a result, the number of tests carried out is limited by the cost and availability of competent engineers and the coverage of the resulting test set can be difficult to evaluate. By automatically generating the test input data and an oracle (to decide the verdict), test cases can be selected to respect criteria whose justification is based on explicit hypotheses, enabling the precise quantification of the level of confidence in the software gained by testing. What is more, the test set can be updated for free for each new version of the software. This is why we are convinced that the reduction of the costs and duration of software verification, as well as the improvement of software quality,

and of quality control, cannot be realised without increased automation of the test process. To this end, we are studying how to implement the automation of the different stages of the test process and have already implemented several prototype tools[7].

A)Terminology :Current research in the field of software testing is characterized by the widespread use of certain methods. The problem can be broken down into three main steps :

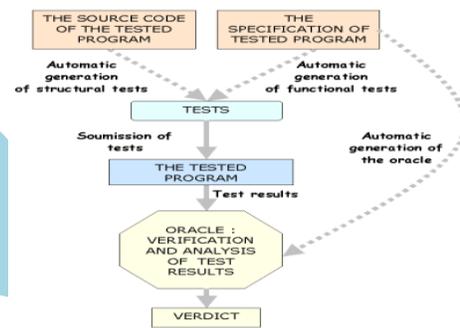


Fig: 1 Automatic Analysis

- Automatic test generation, based either on the program source code, which is called structural testing, or on a formal specification of the tested system, called functional testing.
- Running the tests on the program.
- Automatic analysis of the test results by a program called the oracle.

III WHAT TO TEST

Testing tools can help automate tasks such as product installation, test data creation, GUI interaction, problem detection (consider parsing or polling agents equipped with oracles), defect logging, etc.,without necessarily automating tests in an end-to-end fashion.[3][10].One must keep following popular requirements when thinking of test automation:

- Platform and OS independence
- Data driven capability (Input Data, Output Data, Meta Data)
- Customizable Reporting (DB Access, crystal reports)
- Easy debugging and logging
- Version control friendly – minimum or zero binary files
- Extensible & Customizable (Open APIs to be able to integrate with other tools)
- Common Driver (For example, in the Java development ecosystem, that means Ant or Maven and the popular IDEs). This enables tests to integrate with the developers' workflows.

- Supports unattended test runs for integration with build processes and batch runs. Continuous Integration servers require this.
- Email Notifications (Automated notification on failure or threshold levels). These may be the test runner or tooling that executes it.
- Support distributed execution environment (distributed test bed)
- Distributed application support (distributed SUT)

IV FRAMEWORK APPROACH IN AUTOMATION

A framework is an integrated system that sets the rules of Automation of a specific product. This system integrates the function libraries, test data sources, object details and various reusable modules. These components act as small building blocks which need to be assembled in a regular fashion to represent a business process. Thus, framework provides the basis of test automation and hence simplifying the automation effort[4]. There are various types of frameworks. They are categorized on the basis of the automation component they leverage. These are:

1. Data-driven testing
2. Modularity-driven testing
3. Keyword-driven testing
4. Hybrid testing
5. Model-based testing
6. Regression Testing

V AUTOMATED REGRESSION TESTING

It is any type of software testing that seeks to uncover software errors by partially retesting a modified program. The intent of regression testing is to provide a general assurance that no additional errors were introduced in the process of fixing other problems. "One of the main reasons for regression testing is that it's often extremely difficult for a programmer to figure out how a change in one part of the software will echo in other parts of the software [6].

a) Regression testing is an integral part of the extreme programming software development method. In this method, design documents are replaced by extensive, repeatable, and automated testing of the entire software package at every stage in the software development cycle.

b) Regression testing can be used not only for testing the correctness of a program, but often also for tracking the quality of its output. For instance, in the design of a compiler, regression testing should track the code size, simulation time and time of the test suite cases.

c) Regression testing should be part of a test plan. Regression testing can be automated." Also as a consequence of the introduction of new bugs, program maintenance requires far more system testing per statement written than any other programming. Theoretically, after each fix one must run the entire batch of test cases previously run against the system, to ensure that it has not been damaged in an obscure way. In practice, such regression testing must indeed approximate this theoretical idea, and it is very costly."

d) Regression tests can be broadly categorized as functional tests or unit tests. Functional tests exercise the complete program with various inputs. Unit tests exercise individual

functions, subroutines, or object methods. Both functional testing tools and unit testing tools tend to be third party products that are not part of the compiler suite, and both tend to be automated. Functional tests may be a scripted series of program inputs, possibly even an automated mechanism for controlling mouse movements. Unit tests may be separate functions within the code itself, or driver layer that links to the code without altering the code being tested. Unit testing tools include open source tools like CppUnit and CppTest as well as commercial products like Sword.

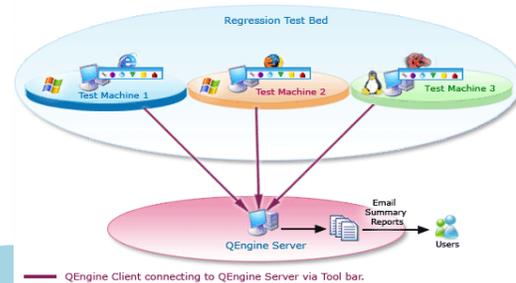


Fig: 2 Regression Testing Architecture

E) Manage Engine Q Engine is a feature rich regression testing tool. The Regression testing capability helps to automate testing of functional, performance and load handling capability of web applications as new functionality is added to the software. In addition to being easy to setup and maintain, QEngine supports a distributed architecture, which can scale to your organization needs and help your development team to produce high quality software.

VI REGRESSION TESTING FEATURES

- **Easy Test Setup and Execution:** Time taken to install the testing software in regression test machine is vastly reduced as it takes only seconds to install the toolbar.
- **Powerful Suite Components:** Provides set of interfaces that enables you to perform fine tuning of the test according to specific operating needs.
- **Easy Test Maintenance:** Object repository to accommodate changes that are made to the application's user interface.
- **Test Selection:** Provision to configure and activate filters to select individual or a group of tests to run in the test machine.
- **Test Scheduling:** Easy-to-use web UI to schedule test suite execution.
- **Distributed Testing:** Provision to run regression tests on multiple machines and different platforms at the same time.
- **Unattended Testing:** Individual and/or group of tests can be executed singly or in parallel from one or many workstations.

VII MANUAL VS AUTOMATED TESTING

- Manual Testing is time consuming.
- There is nothing new to learn when one tests manually.

- People tend to neglect running manual tests.
- None maintains a list of the tests required to be run if they are manual tests.
- Manual Testing is not reusable.
- The effort required is the same each time.
- One cannot reuse a Manual Test.
- Manual Tests provide limited Visibility and have to be repeated by all Stakeholders.
- Only the developer testing the code can see the results.
- Tests have to be repeated by each stakeholder for e.g. Developer, Tech Lead, GM, and Management.
- Manual Testing ends up being an Integration Testing.
- In a typical manual test it is very difficult to test a single unit.
- Scripting facilities are not in manual testing.

Tool Name	Company Name	Latest Version
HP Quick Test Professional	HP	10.5
IBM Rational Functional Tester	IBM Rational	8.1.0.3
Rationalrobot	IBM Rational	2001
Selenium	Open Source Tool	1.0.1
Silk Test	Micro focus	2009
Test Complete	Automated QA	7.52
Test Partner	Micro Focus	6.3
WATIR	Open Source Tool	1.6.5

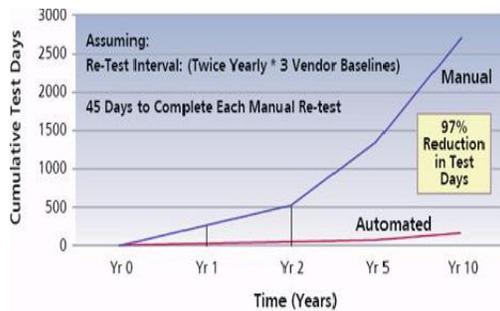


Fig 3. Automated Testing savings over Time

VIII. PROPOSED AUTOMATED TESTING

Automated testing with Quick Test addresses these problems by dramatically speeding up the testing process. You can create tests that check all aspects of your application or Web site, and then run these tests every time your site or application changes [6], [8] .

Effective Benefits of Automated Testing	
Fast	Quick Test runs tests significantly faster than human users.
Reliable	Tests perform precisely the same operations each time they are run, thereby eliminating human error.
Repeatable	You can test how the Web site or application reacts after repeated execution of the same operations.
Programmable	You can program sophisticated tests that bring out hidden information.
Comprehensive	You can build a suite of tests that covers every feature in your Web site or application.
Reusable	You can reuse tests on different versions of a Web site or application, even if the users interface changes.

Table1 Benefits of Automated Testing

IX AUTOMATED TESTING TOOLS

Table 2: Categories of Automated Tools.

X EFFECTIVE TECHNIQUES OF QTP

Quick Test Professional is UI automation software designed for testing Web-based applications running on Microsoft Windows. Like other test automation tools, it works by identifying the objects in the application UI or a web page and performing the desired operations on them (like mouse clicks or keyboard events); it can also be used to capture object properties like name or handler ID etc. To perform these actions, QTP uses a scripting language built on top of VBScript to specify the test procedure, and to manipulate the objects and controls of the application under test. To perform sophisticated actions, users may need to manipulate the underlying VBScript.

A) Plug-ins: QTP offers plug-ins for specific application technologies, such as ActiveX controls, web applications, Visual Basic, Microsoft .NET and others. A limited set of basic plug-ins comes by default. Users need to purchase licenses for others at an additional cost.

B) Record and Playback: Initial development of automated tests with QTP is usually done by record-and-playback. A user's actions are recorded and transposed into comprehensible actions using VBScript. Once recorded, the scripts are editable in either Keyword View or Expert View.

C) Exception Handling: Recovery is the name for exception handling in QTP, with the goal of enabling the tests to continue to run if an unexpected failure occurs. For instance if an application crash occurs and a message dialog appears, QTP can be instructed to attempt to restart the application and continue with the rest of the test cases from there. Because QTP hooks into the memory space of the applications being tested, some exceptions may cause QTP to terminate, and may be unrecoverable.

D) Data Driven Testing: QTP has features to enable users to perform data-driven testing. For example, data can be output to a data table for reuse elsewhere. Data-driven testing is implemented as a Microsoft Excel workbook that can be accessed from within QTP. There are two types of

Data Tables available in QTP: the Global data sheet and the local data sheets. The test steps read data from these data tables in order to (for example) drive variable data into the application under test, and verify the expected result.

XI. DYNAMIC SCENARIO OF AUTOMATED TOOL

1. Ease of use.
2. Presents the test case as a business workflow to the tester (simpler to understand).
3. Uses a real programming language (Microsoft's VBScript) with numerous resources available.
4. Quick Test Pro is significantly easier for a non-technical person to adapt to and create working test cases, compared to Win Runner. Data table integration better and easier to use than Win Runner.
5. Test Run Iterations/Data driving a test is easier and better implementing with Quick Test.
6. Parameterization easier than Win Runner.
7. Can enhance existing Quick Test scripts without the "Application under Test" being available; by using the Active Screen.
8. Better object identification mechanism.
9. QTP supports .NET development environment. XML support, VB Script also.

XII CONCLUSION

Testing is more than just debugging. Testing is not only used to locate defects and correct them. It is also used in validation, verification process, and reliability measurement. Testing is expensive. Automation is a good way to cut down cost and time. Testing efficiency and effectiveness is the criteria for coverage-based testing techniques. New standards are being introduced at international Level. Besides this the actual benefits of adopting reuse is generally observed after long time, therefore use of reuse is ignored by most of the organizations but using software automated testing techniques is much beneficial for software industries. In spite of automated software testing tool is also more effective and useful for tester and developer for implemented and testing to the software in the market.

REFERENCES

- [1] Johannes Samtinger, Software Engineering with Reusable Components, Springer- Verlag, March 1997.
- [2] Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Software. Addison-Wesley, 1995.
- [3] A.Leitner, I.Ciupa, B.Meyer, and M.Howard. Reconciling manual and Automate testing: The auto test experience. In HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences, page 261a, Washington, DC, USA, 2007. IEEE Computer Society.
- [4] C. Boyapati, S. Khurshid, and D. Marinov. Korat: Automated testing based on Java predicates. In ISSTA '02: Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis, pages 123–133, New York, NY, USA, 2002 ACM Press
- [5] V. Basili and R. Selby. Comparing the effectiveness of software testing strategies. IEEE TSE, 13(12):1278–1296,1987
- [6] M. Fewster, D. Graham: Automating Software Testing: Effective use of test execution tools. Addison- Wesley 1999
- [7] T . Xie, D. Notkin, and D. Marinov. Rostra: a framework For detecting redundant object- oriented unit tests. Automated Software Engineering, 2004. Proceedings. 19th International Conference on, pages 196–205, 2004.

- [8] T. Xie, D. Notkin, and D. Marinov. Rostra: a framework for detecting redundant object-oriented unit tests. Automated Software Engineering, 2004. Proceedings. 19thInternational Conference on, pages 196–205, 2004
- [9] IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990), IEEE Computer Soc., Dec. 10, 1990.
- [10] V. Basili and R. Selby. Comparing the effectiveness of software testing strategies. IEEE TSE, 13(12):1278–1296,1987.