Er. RAJENDER BATHLA

DEPTT. OF INFORMATION TECHNOLOGY, HCTM KAITHAL HARYANA (INDIA

dr.bathla@gmail.com

Er. SHALLU BATHLA

DEPTT. OF COMPUTER SC & ENGG S.B.I.E.T, PUNDRI KAITHAL, HARYANA(INDIA)

shallu.miglani@gmail.com

Phase-III: Theme Based Paper

ABSTRACT

•The objective of this research paper is to discuss and emphasis on the Importance of Modern approaches of automated tools with associate to Software Testing & Techniques in the Current Technology. This research paper is focuses on automated software testing using with the modern technology i.e. automated tools in the IT world. The solution of this problem leads to the new innovative approaches in software engineering also associate with automated software testing. These tools are having magnificent and significant role in MNCs
companies, software development organizations and software testing sectors. Software Test Automation is the process of automating the steps of manual test cases using an automation tool or utility to shorten the testing life cycle with respect to time. The objective of software testing is considered to succeed when an error is detached. Software testing having its three major objective i.e. Software Quality improvement (SQI), Verification and validation (V&V), Software Reliability Estimation (SRE

KEYWORDS

•Automated Testing Tools
•Automated Software Testing
•Software testing techniques
•Manual Testing

1. Introduction

Software 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 [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. Let us focuses on objective of software testing.

2. Software Testing Types

There are many ways to conduct software testing, but the most common methods rely on the common steps. There are two basic types of methodical testing.

- Non-Execution-based testing:" The module is reviewed by a proper team"
- Execution based testing: "The module is run against test cases. See fig (2)

2.1 Non Execution based testing

It relies on fault detection strategy. In general, non execution-based code testing is less expensive
than execution – based testing (running test cases) can be extremely time consuming.
Reviews lead to detection of faults earlier in the life cycle. [1], [2].
"Non execution based testing is also known as static testing or static program analysis also.

2.2 Execution Based Testing

In this type of testing, the modules are run against test cases. Following are the two ways of Systematically constructing test data to test a module (functional) or the system performance (non - functional) is unacceptable. [3]

2.2.1 Black Box Testing

Black Box Test design treats the system as a black-box and manual test cases based. So it is a software testing technique whereby the internal workings of the item being tested are not known by the testers. It is also known as functional testing. Other names for black-box testing include: Specification testing, behavioral testing, data -driven testing, functional testing, and input/output-driven testing

2.2.2 White Box Testing

White -box test design allows one to peek inside the box, and it focuses specifically on using internal knowledge of the software to guide the selection of test data.

White-box testing is also known by other names such as Glass-box testing, structural testing, clear box testing, open box testing, Logic-driven testing, and path-oriented testing. [1]

3. Testing Process Stages

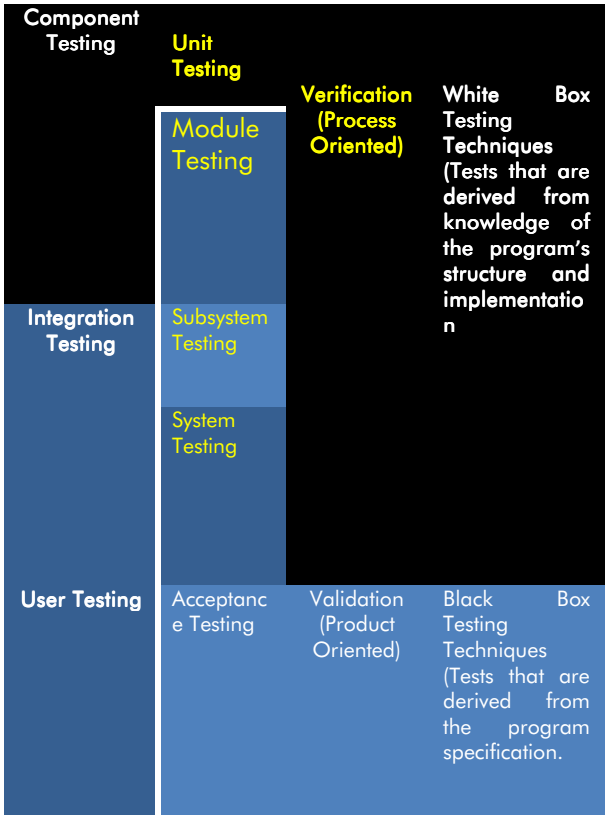

Table (2) Software Testing Techniques

3.1 Unit Testing

Unit testing is code-oriented testing. Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components'

3.2 Module Testing

A module is a collection of dependent components such as an object, class, an abstract data type or some loser collection of procedures and functions. A module encapsulates related components so it can be tested or checked without other system modules.

3.3 Sub-system Testing

This phase involves testing collections of modules, which have been integrated in to sub systems. It is a design-oriented testing and is also known as integration testing.

3.4 System Testing

The sub-systems are integrated to make up the entire system. It is also concerned with validating that the system meets its functional and non-functional requirements. [4].

3.5 Acceptance testing

This is the final stage in the testing process before the system is accepted for operational use. Acceptance testing may also reveal requirement problems where the system facilities do not really meet the user's needs [5]

"Let us see there are many problems if we test to the above mentioned software testing techniques using manual testing rather automated tools".

4. Problems with Manual Testing

With the manual testing we have to write the code or test cases again and again, that is very time consuming. If we update in one unit then it checks again and repeat the all process. More efforts are required in manual testing. **Manual testing** is the process of manually testing software for defects. It requires a tester to play the role of an end user, and use most of all features of the application to ensure correct behavior. To ensure completeness of testing,

the tester often follows a written test plan that leads them through a set of important test cases. [7]

1. Manual Testing is time consuming.
2. There is nothing new to learn when one tests manually.
3. People tend to neglect running manual tests.
4. None maintains a list of the tests required to be run if they are manual tests.
5. Manual Testing is not reusable.
6. The effort required is the same each time.
7. One cannot reuse a Manual Test.
8. Manual Tests provide limited Visibility and have to be repeated by all Stakeholders.
9. Only the developer testing the code can see the results.
10. Tests have to be repeated by each stakeholder for e.g. Developer, Tech Lead, GM, and Management.
11. Manual Testing ends up being an Integration Test.
12. In a typical manual test it is very difficult to test a single unit.
13. Scripting facilities are not in manual testing.

5. Emphasis on Automation Concept

5.1 What is Test Automation?

- Software Test Automation is the process of automating the steps of manual test cases using an automation tool or utility to shorten the testing life cycle with respect to time. [5]
- Automation helps to avoid human errors and also expedite the testing process.
- To implement the Test Automation detailed planning and effort is required.
- Automation saves time and effort which results in reduction of the Test life cycle. [6]

5.2 Why it is Significant?

There is no doubt automated software testing is an activity whose costs are very high. But suppose, once testing tool has purchased for companies or any organization that will be carry and test to thousands of projects. But in the case of Manual Testing, it is not capable for this kind of activities. Manual testing involves a lot of effort, measured in person per month. Using automated testing, with specific tools, this effort can be dramatically reduced and the costs related with testing can decrease. [5]

5.3 Benefits of Automation

- Consistency of Test Execution
- Reducing cycle time of regression test cycle
-Data driven testing
- Repeatability
- Reliability
-Reusability of test wares

Release	Manual Test	Auto Test	Manual Test Cumulative
1	10	10	10
2	10	0	20
3	10	0	30
4	10	0	40
5	10	0	50

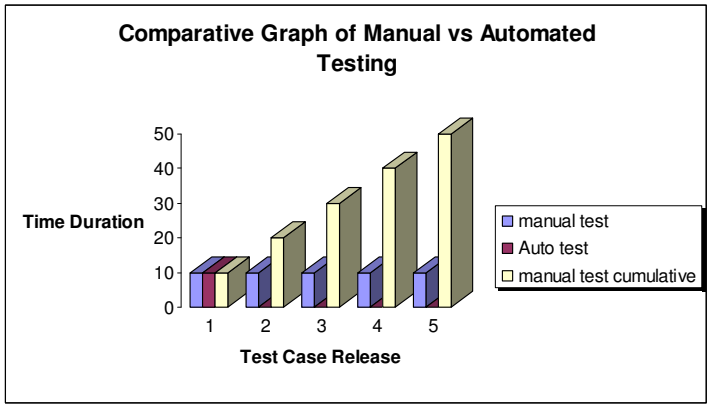

Figure: 1 Comparative Result of Manual Vs Automated Testing

5.4 Innovative approaches of Automated Testing Tools.

- Load Runner
- Win runner
- Silk Test
- QTP
- Rational Rose

5.4.1 Load Runner

Load runner's performance and load testing product by Hewlett-Packard (since it acquired Mercury Interactive in November 2006) for examining system behavior and performance, while generating actual load. Load Runner can emulate hundreds or thousands of concurrent users to put the application through the rigors of real-life user loads, while collecting information from key infrastructure components (Web servers, database servers etc). The results can then be analyzed in detail, to explore the reasons for particular behavior.

5.4.2 Win Runner

It is an automated functional GUI testing tool that allows a user to record and play back UI interactions as test scripts. Win Runner is functional testing software for enterprise IT applications. It captures, verifies and replays user interactions automatically, so you can identify defects and determine whether business processes work as designed.
the Software has some Add-Ins.

- Windows
- C++/C
- Visual Basic
- PowerBuilder
- Forte
- Delphi
- Smalltalk

Automatic Recovery

The Recovery Manager provides an easy-to-use wizard that guides you through the process of defining a recovery scenario. But these facilities are not in manual testing. You can specify one or more operations that enable the test run to continue after an exception event occurs. This functionality is especially useful during unattended test runs, when errors or crashes could interrupt the testing process until manual intervention occurs.

5.4.3 Quick Test Professional

(QTP) is an automated functional Graphical User Interface (GUI) testing tool that allows the automation of user actions on a web or client based computer application.
It is primarily used for functional regression test automation launched by HP (Mercury QTP).

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. [6]

Features of QTP:

- Ease of use.
- Quick Test Pro is significantly easier for a non-technical person to adapt to and create working test cases, compared to Win Runner.

5.4.4 Rational Robo (IBM)

- **Rational Machines** was founded by Paul Levy and Mike Devlin in 1981 to provide tools to expandthe use of modern software engineering practices, particularly explicit modular architecture and iterative development. **Rational** was sold for **US$ 2.1 billion** to IBM on February 20, 2003.Rational Rose is a tool set produced and marketed by Rational Software Corporation (now owned by IBM

  Rose is an operational tool set that uses the Unified Modeling Language (UML) as its means for facilitating the capture of domain semantics and architecture/design intent.
  Various phases and tools used in each phase for software development:

- **Requirement gathering for tool used**: Requisite Pro Designing for tool used: Rational Rose
- **Coding for tool used**: Either manually or by Rational Rose
- **Testing for tool used**: *Purifier, Quantifier, Pure Coverage. Rational Robo*
- **Maintenance for tool used**: Clear Case and Clear Quest.
- **Documentation for tool used**: SODA (Software documentation Automation)
  So there is no doubt with the help of automated tools, testing processes / techniques become much better for software development organization

6. Future Scope of Automated testing tool in the Current Technology

Automated testing trends in the industry suggest that software testing in the future will look very different than it does today. Businesses are leading change

and transformation projects and application development is changing to support it; automated testing approaches also covers quickly rather manual based testing to the current information and communication technology includes e.g.

- Web based applications
- Service Oriented Architectures (SOA)
- Software as a Service
- Wireless technologies
- Mobile technologies

These technologies are growing very fast and giving us quality assurance on the behalf of and good automated testing processes.

7. Conclusion

The Conclusion of this research and review paper is analyze to the manual testing drawback in software testing rather more benefits of automated software testing tools. The enlightened of this modern approaches leads to the new Methodologies of software test automation. The destination of software testing is considered to succeed when an error is detached. Effective Conclusions are given below. Software testing is an art. Most of the testing methods and practices are not very different from 20 years ago. In the current era there are many tools and techniques available to use.

- Good testing also requires a tester's creativity, experience and intuition, together with proper techniques.
- 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.
- Although manual testing is not expensive but is no more effective rather automated testing because automation is a good way to cut down cost and time. Testing efficiency and effectiveness is the criteria for coverage-based testing techniques
- Complete testing is infeasible. Complexity is the root of the problem. At some point, software testing has to be stopped and product has to be shipped. The stopping time can be decided by the trade-off of time and budget. Or if the reliability estimate of the software product meets requirement.
- Testing may not be the most effective method to improve software quality. Alternative methods, such as inspection, and clean-room engineering, may be even better.

References

1. Burnstein, Ilene, Practical Software Testing. Springer Verlag New York Inc., USA, 2003
2. Beizer, B. Software Testing Techniques. Second Edition,Van Nostrand Reinhold Company Limited,1990
3. Gill, G. and C. Kemerer. "Cyclomatic Complexity Density and Software Maintenance Productivity". IEEE Transactions on Software Engineering, December 1991.
4. Gerald, D. Everett et al. Software Testing: Testing Across the Entire Software Development Life Cycle. John Wiley & Sons, Inc. Publication, New Jersey, 2007.
5. T. Chow, "Testing Software Design Modeled by of the IEEE International Conference on Automated Software Engineering, Montreal, Quebec,
6. Mosley, Daniel (2002). Just Enough Software Test Automation.
7. A. Leitner, I. Ciupa, B. Meyer, and M.Howard. Reconciling manual and automated testing: The auto test experience. In HICSS 07: Proceedings of the 40th Annual Hawaii International Conference on USA, 2007. IEEE Computer Society.

GJEIS

Volume-1 Issue-1

January 2009-June 2009

Phase-III: Theme Based Paper

Phase-III: Theme Based Paper