

Software Testing Research: Emerging Trends in Tools & Techniques, Challenges and Predictions

Shubham Kumar^{1,2}, Akansha Garg¹, Amanpreet Singh^{1,3}

B. Tech Students, Department of Computer Science Engineering, Dronacharya College of Engineering,
Greater Noida, India¹

Founder, Websoles.com, India²

Co-founder, Websoles.com, India³

ABSTRACT: Software testing is the process of testing the software products. Effective software testing will contribute to the delivery of higher quality software products, more satisfied users, lower maintenance costs, more accurate, and reliable results. In other words, Software testing is an important discipline and consumes significant, amount of effort. We can also discuss many points that is who should do the testing? What should we test? What are the latest Tools used? It is more important because Testing is the process of executing a program with the intent of finding errors. One of the major problems with software testing area is how to get a suitable set of cases to test a software system with more accurately. Day after day life problems became complex and the Software developed to solve those problems having more complexity. Software testing is an important area of research and a lot of development has been made in this field. In this paper, testing techniques and tools have been described. This study aims to compare the main features of different scripting techniques used in process of automating the execution phase in software testing process. Software testing is gaining more and more importance in the future. Some typical latest researches have been summarized.

KEYWORDS: Test Process Analysis , Level of Testing, Test Approach , Software testing, Software testing strategies, Testing tools, Test plans, Software testing principles, Research orientation.

I. INTRODUCTION

Software Testing is a process of finding errors while executing a program so that we get a zero defect software. It is aimed at evaluating the capability or usability of a program. It is also the process of evaluating the developed system to assess the quality of the final product. Unfortunately, software-testing process is expensive and consumes a lot of time through software development life cycle. As software systems grow, manual software testing becomes more and more difficult. Though a lot of advancements have been done in formal methods and verification techniques, still we need software to be fully tested before it could be handled to the customer side. Software testing plays an important role in the Software Development Life Cycle. Model-based testing has gained a lot of prominence in recent times with organizations looking for automation of test case design and execution in lieu of a manual testing process. Model-based testing techniques help generate numerous test artifacts such as test cases, test sequences, executable test scripts and requirement traceability matrix. In model-based software engineering, models can broadly be classified into two categories: 1. Static models and 2. Dynamic models Static models represent the structural aspects of the system (e.g., class diagrams). On the other hand, dynamic models capture the behavioral aspects of the system (e.g., activity diagrams). In this thesis, techniques to generate test sequences from a variety of dynamic models are explored. Test sequence generation algorithms are developed for two different classes of models. For one algorithm, the subject chosen is UML sequence diagrams and activity diagrams. Another algorithm is developed for generating test sequences

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

from business process models. In this approach, business rules are externalized from business process models and then test sequences are generated. In summary, the main contributions of this research are novel modeling and test sequence generation approaches to generate test sequences from different classes of models. Also for business process models, business rules are externalized from business process models as this could aid enterprise organizations which use business process models with embedded rules. In addition to test sequence generation algorithms, an analysis framework for model-based testing has also been developed. It helps assess the type of test artifacts that can be generated from UML models at different levels of granularity.

II. SOFTWARE TESTING IS ALL ABOUT AUTOMATION

Automation is just like every other technology in that it takes effort to be successful. It's another strategy that can be incorporated into the software testing life cycle, but issues come from testers figuring out how they can build it into a process and determining what solutions or tools are right for their company. Over the past few years, testing went from 100% manual to incorporating automation so tests could be fast and repeatable. With mobile, agile and DevOps as driving forces, the payoff and ROI for automation that was promised years ago is finally happening, according to Mobile Labs vice president of mobility solutions Dan McFall.

Since businesses are moving to faster releases, the time to market has changed. According to Anand Kamat, group program manager at Microsoft, getting a software idea to become reality in a short timeframe puts pressure on organizations. Although the focus is on automation (and rightfully so), manual testing is still relevant and takes many different forms in agile, said Kamat. Developers are continuing to incorporate more automated tests, with "test early, test often" still the guiding principle.

2.1 The risks of automated testing

Automation isn't a one-size-fits-all solution. Companies have to complement test automation metrics with data from manual testing, exploratory testing, user acceptance testing, and testing in real-world customer environments to form a comprehensive view about quality of the product, according to Anand Kamat, group program manager at Microsoft. He suggests considering the following risks before getting started:

- ✚ Most often automation does not cover 100% of use cases, but having a consistent pass rate of 100% can give you a sense of "false confidence" that is fatal.
- ✚ Code coverage is not a reliable metric for ensuring end-to-end quality, but is often used as a measure to gauge the effectiveness of test automation.
- ✚ For connected applications involving multiple components, your "one-box" setup is not a real-world scenario. If you are not testing in an "integration environment," you are not testing with right dependencies.
- ✚ With frequent changes in user experience, ROI on UI Automation might be limited for multi-channel applications. Having a services/API testing strategy combined with exploratory testing might be a better alternative.

2.2 Speed over quality

Because of the rise of automated tests; the need for speed continues to stay on companies' minds. Matt Brayley-Berger, worldwide product marketing manager for HPE, said that his company always likes to ask the question, "Would you like to have better quality or faster speed?" They pondered the question, eventually reaching a consensus that organizations are finding ways to remove a lot of barriers that would have already lowered quality, according to him. This means they can still focus on the speed of release without sacrificing quality. And, if testers became more technical and had to work with development a little more closely, it forces the DevOps conversation sooner. "It's not the solution, but maybe that behavior is creating an environment to have more productive conversations with evolving testing and evolving quality," said Brayley-Berger.

The shortcuts that take place during the development process can take a one-month turnaround to a one-week timeframe by simply reducing the amount of testing that is being done. According to Walter Capitani, a product

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

manager at Rogue Wave, this is what companies are doing. He said they take shortcuts in testing by reducing the tests or by doing things after they've finished the release because they figure if they find a quality problem, they can always "patch it later."

III. SOFTWARE TESTING IS EVOLVING RAPIDLY

The introduction of new architectures such as service-oriented architecture (SOA), increase in virtualization and inclination towards cloud has shaken up the industry and has compelled testers to consider new aspects. Additionally, focus on DevOps has dramatically changed the face of IT.

There are several important trends in software testing world that will alter the landscape that testers find themselves in today:

- ✚ Mobile application testing (Rapid growth in Mobile testing)
- ✚ Cloud-based Testing (Virtualization and Cloud Computing)
- ✚ Testing in the Agile Development Environment
- ✚ Context-driven test
- ✚ Security Testing
- ✚ Crowd sourced testing
- ✚ Proliferation of open-source testing tools
- ✚ Tester Certification

3.1 Mobility Application Testing (Rapid growth in Mobile testing):

With the surge of mobile devices in the market, there is a significant increase in mobile testing activity, including functionality, performance and security of mobile applications and devices. Some of the applications today are using the power of cloud to communicate with the ERP systems. In such situations, data security is of paramount importance. The multitude OS used on the handsets adds to the challenge calling for functionality testing and integration testing.

Mobile devices have become a major part of many Mobile App Testing organizations' digital development strategy, but many businesses lack the testing specialists, tools, and consistent methodology needed for effective mobile development. Mobile market growth is nothing new; the mobile industry has been skyrocketing for several years, as consumers spend more time on multiple devices and consume mobile-only content. According to the World Quality Report, 55% of organizations now test mobile apps, up from 31% last year. Yet, 56% lack effective testing procedures. Security also remains a key consideration for mobile testing. With a growing number of devices interacting in the Internet of Things, safety must be a top concern in QA without compromising the very mobility and wireless access that enable mobile devices. While increased mobile usage remains the future for consumers, businesses have yet to develop effective testing procedures that can achieve effective test coverage without falling prey to the accelerating development lifecycle.

3.2 Cloud-based Testing (Virtualization and Cloud Computing):

As cloud computing becomes an increasing part of the IT mainstream with about 26% of software applications to be hosted on cloud in 2015, IT professionals and experts see a positive growth in the adoption of cloud based testing. For the simple reason, cloud infrastructure offers a convenient solution unlike other test environments which can be difficult and expensive to build and maintain. There are concerns related to data security and performance in cloud based environments which organizations view as a medium term challenge that will be soon addressed. Cloud testing is also flexible, allowing businesses to scale up or down as part of a dynamic testing strategy.

3.3 Testing in the Agile Development Environment:

Companies are working hard to build a holistic testing approach that fits with the agile development methodology and

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

brings in right test tools. Continuous testing is something that we will be seeing a lot as companies today are more focused on getting into the delivery phase without having to wait for long. Testing in production is one approach that will be commonly used by testers in order to provide a constant flow of updates on how the software is shaping up with respect to its features and value to business. Agile methods are in the mainstream. They offer both challenges and opportunities for testers. Agile does not mean one thing, but a collection of many related things. Integration of testing best practices into Agile methodologies remains a challenge. Agile Scrum is another development that is changing the manner in which testers operate. This is a flexible, holistic product development strategy that encourages teams to self-organize and communicate regularly. In other words, testers work closely with developers and obtain early involvement and feedback on a project.

3.4 Testing in Agile Methodology

3.4.1 Context-driven Testing

A one-size fits all approach is not something that we can rely on when it comes to testing of software applications. There is a need to be adaptable in the event the context changes. This has made context-driven software testing a trend that we expect will emerge strongly in the coming years. According to software testing leaders in the industry, the most successful testers in the future will be the ones that can bring the most skills to the table for any given context or business situation.

3.4.2 Security Testing

The increased connectivity of data and information systems along with the rise of cloud based computing and the phenomenon of Internet of Things taking center stage in the IT industry, security testing of software applications is going to be of top priority.

4.3 Crowd Sourced Software testing

This form of testing is slowly gaining popularity as it helps organizations thrift its budget for software development. It is helpful in situations wherein the testing team may not have all the resources to test the software in different environments and in different scenarios.

All these trends are testimony of the fact that the software testing industry is evolving at an enormous pace. What will be interesting for us to watch is how this industry is going to adapt itself to the growing demands of technology and innovation in the IT sector.

3.5 Open-source testing tools

Agile methods have spun-off many useful open-source test tools like X-Unit, Selenium etc. Test automation skills are essential to take advantage of these tools. Integrated development environments often include related or similar tools.

IV. LATEST TRENDS IN SOFTWARE TESTING

Blame it on globalization or blame it on the advent of new platforms and technology. Whatever the reason, businesses all around the world are rapidly changing and QA is not immune to it. Year on year, we are seeing new trends in testing that have changed the way we test software. Here are the top 7 trends seen in 2013-2014:

- 1. Integrating QA through agility and TCoE:** QA is seeing greater integration into business through setting up agile testing teams and Testing Centers of Excellence (TCoE). Though agile development teams have been around for a long time, agility in testing is still nascent. With the continuous pressure to quickly deliver software, businesses are investing time and money into setting up a TCoE with the objective of reducing CoQ, increasing test effectiveness and generating more ROI out of testing. From 2011 to 2014, the number of operational TCoEs has increased from 4% to 19% and is expected to increase further in the future.
- 2. Greater focus on new technology:** New technologies like SOA, cloud and mobile testing are on the rise. According to 2013-2014 world quality report, mobile testing has seen a rapid rise from 31% in 2012 to 55% in

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016




2013. Yet 56% lack comprehensive mobile testing procedures. By 2015, almost 36% of software will be hosted in the cloud but businesses still lack the necessary infrastructure for cloud testing. These might cause businesses to opt for TaaS (Testing as a service) options.

3. **Higher Automation Levels:**With agile testing teams and growing number of TCoEs, and due to high pressure to reduce the time-to-market, testing teams are trying to adopt automation wherever possible. This is not just in regression but also in unit testing and load testing.
4. **Greater focus on security:**System robustness and security has always been a top priority but with growth in social media and mobility and need for software that can be integrated to multiple platforms, systems are becoming more vulnerable. There is a pressing need to ensure enhanced security, particularly in applications handling sensitive data. This is causing QA to focus more on security testing.
5. **Context driven testing:**Another effect of the spread of social media, localization and mobility is the result of devices that, though diverse, are still tightly integrated in terms of software and usage. It is challenging for businesses to maintain central hubs of hardware, middleware and test environments necessary to comprehensively test them. This has caused context driven testing to become more popular as it ensures more testing coverage from diverse angles. It is expected that this will impact skill development among testers as there will be more demand for testers with exposure to different contexts.
6. **Continuous Integration testing:**Under this methodology, testing is done in small increments in a production like environment with code integration happening at frequent intervals. This kind of testing helps the tester to detect problems early, measure the effectiveness of a change and determine whether it actually meets end-user expectations.
7. **Independent Software testing:**With increased focus on QA, many businesses are relying on specialist QA organizations to provide testing services. This is largely due to the expertise that specialist QA organizations bring to the table, including TCoE capability. Hence partnering with them helps the business to circumvent the pain of finding skilled QA resources and setting up a mature QA, both in terms of process and technology. A report from business research firm Nelson Hall states that the independent testing sector will see an annual growth of 9.5% for the next few years.

More and more businesses around the world are awakening to the need to have a well-designed and futuristic quality assurance process to maintain their edge in the industry. Partnering with a specialist QA service can help you readily adapt to the latest trends in testing.

V. SOFTWARE TESTING TOOLS

Now days we can get lots of **Software Testing Tools** in the market. Selection of tools is totally based on the project requirements & commercial (Proprietary/Commercial tools) or free tools (Open Source Tools) you are interested. Off Course, free Testing Tools may have some limitation in the features list of the product, so it's totally based on what are you looking for & is that your requirement fulfill in free version or go for paid Software Testing Tools. The tools are divided into different categories as follows:

-  Test Management tools
-  Functional Testing Tools
-  Load Testing Tools

Here you can get a most popular free and paid **Testing Tools list** used in the actual testing of the software application.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

5.1 Open Source Tools

a) Test Management tools

TET (Test Environment Toolkit)

The goal behind creating the Test Environment Toolkit (TET) was to produce a test driver that accommodated the then current and anticipated future testing needs of the test development community. To achieve this goal, input from a wide sample of the community was used for the specification and development of TET's functionality and interfaces.

Test Manager

The Test Manager is an automated software testing tool is used in day to days testing activities. The Java programming language is used to develop this tool. Such Test Management tools are used to facilitate regular Software Development activities, automate & manage the testing activities. Currently Test Manager 2.1.0 is ready for download.

RTH is called as "Requirements and Testing Hub". This is a open source test management tool where you can use as requirement management tool along with this it also provides the bug tracking facilities.

b) Functional Testing Tools

Selenium

Soapui

Watir

HTTP::Recorder

WET

WebInject

c) Load Testing Tools

Jmeter

FunkLoad

5.2 Proprietary/Commercial tools

a) Test Management tools

HP Quality Center/ALM

QAComplete

T-Plan Professional

Automated Test Designer (ATD)

Testuff

SpiraTest

TestLog

b) Functional Testing Tools

QuickTest Pro

Rational Robot

Sahi

SoapTest

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

- ✚ Badboy
- ✚ Test Complete
- ✚ QA Wizard
- ✚ Netvantage Functional Tester
- ✚ PesterCat

c) Load Testing Tools

- ✚ WebLOAD Professional
- ✚ HP LoadRunner
- ✚ LoadStorm
- ✚ NeoLoad
- ✚ Loadtracer
- ✚ Forecast
- ✚ ANTS – Advanced .NET Testing System

5.3 Automated GUI Testing Tools 5 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.

5.3.1 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.

Commercial

➤ [eggPlant](#)

EggPlant is a GUI test automation tool for professional software applications and enterprise teams. It can be used to

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

automate different application types, such as .NET, Java and Flash applications.

5.3.2 Unit Testing Frameworks 9 Tools

See below for a list of popular unit testing frameworks and tools for major platforms and programming languages. These frameworks can be used by programmers to test specific functionality in libraries and applications. Unit tests can then be used to automatically test new versions and builds as part of an automated build system or deployment process.

➤ [NUnit](#)

NUnit is a unit-testing framework for all .Net languages. It was initially ported from JUnit to .NET and has been redesigned to take advantage of many .NET language features.

.NET, Open Source

➤ [xUnit.net](#)

xUnit.net is a community-focused unit testing tool for the .NET Framework written by the original inventor of NUnit. xUnit.net is the latest technology for unit testing C#, F#, VB.NET and other .NET languages.

Python, Open Source

➤ [PyUnit / unittest](#)

The Python unit testing framework, sometimes referred to as "PyUnit", is a Python language version of JUnit. It's part of the Python framework and supports test automation, sharing of setup and shutdown code for tests and various other features. Used by :Java, Open Source

➤ [JUnit](#)

JUnit is a simple unit testing framework to write repeatable tests in Java. JUnit has been important in the development of test-driven development and is one of the standard testing frameworks for Java developers.

Used by :Java, Open Source

➤ [TestNG](#)

TestNG is a Java testing framework inspired by JUnit and NUnit and introduces some new functionality that make it more powerful and easier to use. TestNG is designed to cover multiple categories of tests, including unit, functional, end-to-end and integration tests. Used by :PHP, Open Source

➤ [PHPUnit](#)

PHPUnit is a popular framework for unit testing in PHP projects. It provides both a framework that makes it easier to write tests as well as the functionality to easily run, execute and analyze tests and results. Used by :PHP, Open Source

➤ [Symfony Lime](#)

Lime is a unit and functional testing framework built for the popular Symfony PHP web application framework. The framework is designed to have readable output from tests, including color formatting, by following the Test Anything Protocol. Used by :Ruby, Open Source

➤ [Test::Unit](#)

Ruby comes with its own standard unit testing framework as part of the Test::Unit namespace and can be used to define basic pass/fail tests and group tests. The framework also comes with tools to run single or whole groups of tests. Used by :Ruby, Open Source

➤ [RSpec](#)

RSpec is a testing tool for the Ruby programming language. Born under the banner of behavior-driven development, it is designed to make test-driven development more productive and enjoyable.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

VI. CONCLUSION AND DISCUSSION

A Plethora of opportunities are being created for software testing with each passing day and this would certainly help the Indian IT firms involved in software testing. Verticals such as Cloud Computing and [mobile application development-commerce development](#) are going to provide positive impetus to both the development and testing fraternities. It's time for us to gear up to meet the challenges and taste the fruits of software testing market in 2016.

Considering the above software testing trends, statistics and projections, we can confidently predict a bright future for quality assurance and software testing domain, especially in the areas of automation testing, performance testing and security testing. Software testers need to be prepared and be ready to grab the emerging opportunities in the 13 billion dollar software testing industry

REFERENCES

- [1]. Lessons Learned in Software Testing, by C. Kaner, J. Bach, and B. Pettichord
- [2]. Testing Computer Software, by C. Kaner, J. Falk, and H. Nguyen
- [3]. Effective Software Testing, by E. Dustin [4]. Software testing, by Ron Patton
- [5]. Software engineering, by Roger Pressman
- [6] K.K.Aggarwal[7]www.engpaper.com
- [8].<http://www.people.engr.ncsu.edu/txie/testingresearchsurvey.htm>
- [9].www.cs.cmu.edu/luluo/Courses/17939Report.pdf
- [10].www.findwhitepapers.com
- [11].www.scribd.com
- [12] people.engr.ncsu.edu/txie/testingresearchsurvey.htm
- [13] inpressco.com/wp-content/uploads/2014/07/Paper122368-2372.pdf
- [14] www.cs.cmu.edu/~luluo/Courses/17939Report.pdf
- [15] ijcsi.org/papers/IJCSI-11-2-2-120-123.pdf
- [16]istqbexamcertification.com/whatisintegrationtesting/
- [17]en.wikipedia.org/wiki/Integration_testing
- [18]www.ijcsmc.com/docs/papers/June2014/V3I6201499a11.pdf
- [19]www.cs.cmu.edu/~luluo/Courses/17939Report.pdf
- [20]inpressco.com/wp-content/uploads/2014/07/Paper122368-2372.pdf
- [21]istqbexamcertification.com/whatisystemtesting/
- [22]en.wikipedia.org/wiki/System_testing
- [23]people.engr.ncsu.edu/txie/testingresearchsurvey.htm
- [24]www.engpaper.com
- [25]www.people.engr.ncsu.edu/txie/testingresearchsurvey.htm
- [26]www.cs.cmu.edu/luluo/Courses/17939Report.pdf
- [27] www.findwhitepapers.com
- [28] www.scribd.com