Utilization and Deployment of Software Testing Tools and Techniques

Naw Tin Tin Yu
University of Computer Studies (Hpa-an), Kayin State, Myanmar

Abstract: Software testing is a process of executing a program or application with the intent of finding the software bugs. Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. Software testing tool is used in day to days testing activities.In this paper, testing tools and techniques have been described. Software testing is gaining more and more importance in the future.

Keywords: *Software testing, Tools, Techniques.*

I. INTRODUCTION

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use [2]. Testing is an important technique to check and control software quality. It is partof almost any software development project. Time and resources devoted to testingmay vary from 30% up to 70% of a project's resources. Yet, testing is often an underexposedand underestimated part of software development. Also testing are rather low compared with the importance of the subjectin real-life software projects. The course on testing techniques is intended to be a steptowards bridging this gap.Testing techniques deals with a number of topics related to software testing. The aimis that the student will get insight into some of the problems of software testing andthat he gets familiar with a couple of solutions to these problems. Emphasis ison techniques for testing technical software systems, such as communication software,control systems and embedded software. Established testing techniques as well as newdevelopments will be presented. In the latter category there will be a strong focus onthe use of formal methods in the software testing process. It is impossible to give acomplete view of all possible testing techniques and processes in a single course. Testing techniques is a reasonable compromise between giving a completeoverview of the broad _end of testing and dealing thoroughly with distinct testingtechniques without being too super-cial. [13]

This paper is organized in seven sections: In section 1.Introduction; 2. Literature review; 3.Testing tools; 4. Advanced Software Testing Tools; 5. Testing Techniques; 6.Advanced Software Testing Techniques; 7. Conclusion.

II. LITERATURE REVIEW

Software testing is a very broad area, which involves many other technical and non-technical areas, such asspecification, design and implementation, maintenance, process and management issues in softwareengineering. Our study focuses on the state of the art in testing techniques, as well as the latest techniques which representing the future direction of this area. Software testing techniques are based in an amalgam of methods drawn from graphtheory, programming language, reliability assessment, reliable-testing theory, etc.[12]. Software testing is very important because of the following reasons: Software testing is really required to point out the defects and errors that were made during the development phases. It’s essential since it makes sure of the Customer’s reliability and their satisfaction in the application. It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence. (Know more about Software Quality). Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results. Testing is required for an effective performance of software application or product. It’s important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It’s required to stay in the business. [5]Testing improves the quality of the final product because most of the errors found get corrected before the software is released into production. Test tool selection is a very important part of test automation. This requires the study of the scope of testingand 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: Scripting interface; Facility to give valid and invalid input; Result comparison;To give the verdict.[15]

III. TESTING TOOLS

There are many testing tools available that are useful in several places while testing software product. These tools canbe categorized as static testing and dynamic testing tools.

A. Static Testing Tools

These tools test the software without executing it; rather, they are concerned with analyzing the code or documentation for syntax checking, consistency, etc. Static testing can be manual or even automated with the use of static analysis tools. Static analysis tools examine the source code of program and highlight the statements with wrong syntax, undefined symbols or variables, use of uninitialized variables, and so on. They also check for flaws in the logic flow of the program. Static analysis tools are generally used by developers as part of the development and component testing process. The key aspect is that the code is not executed or run but the tool itself is executed, and the source code we are interested in is the input data to the tool. These tools are mostly used by developers.Static analysis tools are an extension of compiler technology – in fact some compilers do offer static analysis features. It is worth checking what is available from existing compilers or development environments before looking at purchasing a more sophisticated static analysis tool. Other than software code, static analysis can also be carried out on things like, static analysis of requirements or static analysis of websites. Static analysis tools for code can help the developers to understand the structure of the code, and can also be used to enforce coding standards. [16] Static Testing Techniques provide a powerful way to improve the quality and productivity of software development by assisting engineers to recognize and fix their own defects early in the software development process. In this software is tested without executing the code by doing Review, Walk Through, Inspection or Analysis etc. Static Testing may be conducted manually or through the use of various software testing tools. It starts early in the Software Development Life Cycle and so it is done during the Verification Process. Static testing is not magic and it should not be considered a replacement for Dynamic Testing**,** but all software organizations should consider using reviews in all major aspects of their work including requirements, design, implementation, testing, and maintenance.Types of defects that are easier to find during static testing are: deviations from standards, missing requirements, design defects, non-maintainable code and inconsistent interface specifications. [17] Static Testing is type of testing in which the code is not executed. It can be done manually or by a set of tools. This type of testing checks the code, requirement documents and design documents and puts review comments on the work document. When the software is non –operational and inactive, we perform security testing to analyze the software in non-runtime environment. With static testing, we try to find out the errors, code flaws and potentially malicious code in the software application. It starts earlier in development life cycle and hence it is also called verification testing. Static testing can be done on work documents like requirement specifications, design documents, source code, test plans, test scripts and test cases, web page content.[18]

B. Dynamic Testing Tools

These tools interact with the software while execution and help the testers by providing useful information about the program at different events. This information may include the number of times some particular statements is executed, whether all the branches of decision point have been exercised, minimum and maximum values of variables, and so on. While performing testing with automated tools, the following points should be noted.Clear and reasonable expectations should be established in order to know what can and what cannot be accomplished with automated testing in the organization.There should be a clear understanding of the requirements that should be met in order to achieve successful automated testing. This requires the following consideration. The organization should have detailed, reusable test cases, which contain exact expected results and a stand-alone test environment with a restorable database: Technical personnel to use the tools effectively; An effective manual testing process, which must exist before automation begins.Testing tool should be cost-effective. It should involve minimum technical personnel and should ensure that test cases developed for manual testing are also useful for automated testing.Select a tool that allows implementation of automated testing in a way that conforms to the specified long-term testing strategy.Many automated tools are available for performing the testing process in an effective and efficient manner. Automated tools like Mothora are used to design test cases, evaluate their adequacy, verify the correctness of input and output, find and remove the errors, and control and summarize the test. Similarly, Bug Trapper is used to perform white box testing. This tool traces the path of execution and captures the bug along with the path of execution and the different input values that resulted in the error[6].Dynamic analysis tools are 'dynamic' because they require the code to be in a running state. They are 'analysis' rather than 'testing' tools because they analyze what is happening 'behind the scenes' that is in the code while the software is running (whether being executed with test cases or being used in operation). Eventually when our computer's response time gets slower and slower, but it get improved after re-booting, this may be because of the 'memory leak', where the programs do not correctly release blocks of memory back to the operating system. Sooner or later the system will run out of memory completely and stop. Hence, rebooting restores all of the memory that was lost, so the performance of the system is now restored to its normal state. These tools would typically be used by developers in component testing and component integration testing, e.g. when testing middleware, when testing security or when looking for robustness defects. Dynamic analysis for websites is to check whether each link does actually link to something else. The tool does not know if you have linked to the correct page, but at least it can find dead links, which may be helpful. [16] In Dynamic Testing Technique software is executed using a set of input values and its output is then examined and compared to what is expected. Dynamic execution is applied as a technique to detect defects and to determine quality attributes of the code. Dynamic Testing and Static Testing are complementary methods, as they tend to find different types of defects effectively and efficiently. But as it does not start early in the Software Development Life Cycle hence it definitely increases the cost of fixing defects. It is done during Validation Process evaluating the finished product. [17] Dynamic testing is done when the code is in operation mode. Dynamic testing is performed in runtime environment. When the code being executed is input with a value, the result or the output of the code is checked and compared with the expected output. With this we can observe the functional behavior of the software, monitor the system memory, CPU response time, performance of the system. Dynamic testing is also known as validation testing, evaluating the finished product. Dynamic testing is of two types: Functional Testing and Non-functional testing. [18]

C. Utilization of Static and Dynamic Testing Tools

Utilization of static analysis tools are: To calculate metrics such as cyclomatic complexity or nesting levels (which can help to identify where more testing may be needed due to increased risk); To enforce coding standards; To analyze structures and dependencies; Help in code understanding; To identify anomalies or defects in the code.Review is typically used to find and eliminate errors or ambiguities in documents such as requirements, design, test cases, etc. The code written by developers are analyzed (usually by tools) for structural defects that may lead to defects. Utilization of dynamic analysis tools are: To detect memory leaks; To identify pointer arithmetic errors such as null pointers; To identify time dependencies.

D. Advantages and Disadvantages Of Static Testing Tools

ADVANTAGES	DISADVANTAGES
It can find weaknesses in the code at the exact location.	It is time consuming if conducted manually.
It can be conducted by trained software assurance	Automated tools do not support all programming

developers who fully understand the code.	languages.
It allows a quicker turn around for fixes.	Automated tools produce false positives and false negatives.
It is relatively fast if automated tools are used.	There are not enough trained personnel to thoroughly conduct static code analysis.
Automated tools can scan the entire code base.	Automated tools can provide a false sense of security that everything is being addressed.
Automated tools can provide mitigation recommendations, reducing the research time.	Automated tools only as good as the rules they are using to scan with.
It permits weaknesses to be found earlier in the development life cycle, reducing the cost to fix.	It does not find vulnerabilities introduced in the runtime environment.
Since static testing can start early in the life cycle, early feedback on quality issues can be established. By detecting defects at an early stage, rework costs are most often relatively low. Since rework effort is substantially reduced, development productivity figures are likely to increase. The evaluation by a team has the additional advantage that there is an exchange of information between the participants. Static tests contribute to an increased awareness of quality issues.	

E. Advantages and Disadvantages of Dynamic Testing Tools

ADVANTAGES	DISADVANTAGES
It identifies vulnerabilities in a runtime environment.	Automated tools provide a false sense of security that everything is being addressed.
Automated tools provide flexibility on what to scan for.	Automated tools produce false positives and false negatives.
It allows for analysis of applications in which you do not have access to the actual code.	Automated tools are only as good as the rules they are using to scan with.
It identifies vulnerabilities that might have been false negatives in the static code analysis.	There are not enough trained personnel to thoroughly conduct dynamic code analysis [as with static analysis].
It permits you to validate static code analysis findings.	It is more difficult to trace the vulnerability back to the exact location in the code, taking longer to fix the problem.
It can be conducted against any application.	It is a time consuming task because its aim is to execute the application or software and as a result more test cases are needed to execute.
It can always find errors that static testing cannot find and that is the reason why it is considered as high level exercise.	It is not done early in the software life cycle and hence bugs fixed in later stages result in more cost.
Executing the software leads to the chances of finding more bugs in the application, so it ensures error free software to some extent.	It requires more man power to complete the task

F. Difference Between Static Testing and Dynamic Testing

STATIC TESTING	DYNAMIC TESTING
Static Testing is white box testing which is done at early stage if development life cycle. It is more cost effective than dynamic testing	Dynamic Testing on the other hand is done at the later stage of development lifecycle.
Static testing has more statement coverage than dynamic testing in shorter time	Dynamic Testing has less statement stage because it is covers limited area of code
It is done before code deployment	It is done after code deployment
It is performed in Verification Stage	It is done in Validation Stage
This type of testing is done without the execution of code.	This type of execution is done with the execution of code.
Static testing gives assessment of code as well as documentation.	Dynamic Testing gives bottlenecks of the software system.
In Static Testing techniques a checklist is prepared for testing process	In Dynamic Testing technique the test cases are executed.
Static Testing Methods include Walkthroughs, code review.	Dynamic testing involves functional and nonfunctional testing
It reviews the requirement documents, design documents initially	Testing the functionality of the different page.
Checking the GUI of the application	Checking the checkout process and payment methods.
Checking the database structure of the application.	Testing the interfaces between different pages.

IV. ADVANCED SOFTWARE TESTING TOOLS

There are five advanced software testing tools. They are Free software testing tools gnitset ecafretni resu lacihparG ; daoL ; sloot gnitset ytiruceS ;sloot gnitsetskrowemarf gnitset tinU ; nI . oba ,sloot gnitset erawtfos decnavdaut security testing tools describe: [7]

Security Testing Tools

Security Testing Tools areAddress Sanitizer; American fuzzy lop (fuzzer); Jtest; Metasploit Project; Nmap; Parasoft C/C++test; SOAtest.[8]

***A. Address Sanitizer* (or ASan)**

Address Sanitizer (or ASan) is an open source programming tool by Google that detects memory corruption bugs such as buffer overflows or accesses to a dangling pointer (use-after-free). Address Sanitizer is based on compiler instrumentation and directly-mapped shadow memory. Address Sanitizer is currently implemented in Clang (starting from version 3.1) and GCC (starting from version 4.8). On average, the instrumentation increases processing time by about 73% and memory usage by 340%.[7]

B. American Fuzzy Lop

American fuzzy lopis a fuzzer that employs genetic algorithms in order to efficiently increase code coverage of the test cases. So far it helped in detection of significant software bugs in dozens of major free software projects, including X.Org Server, PHP, OpenSSL, pngcrush, bash, Firefox, BIND and Qt.American fuzzy lop's source code is open and is hosted by Michał Zalewski on his website. The program attracts a relatively large and active community - as of 27 Aug 2015, its main mailing list gathers 392 users and developers and generates an average of over 100 posts per month Its name is a reference to a breed of rabbit, American Fuzzy Lop. This article relies too much on references to primary sources. Please improve this by adding secondary or tertiary sources. [8]

C. Jtest

Jtest is an automated Java software testing and static analysis product that is made by Parasoft. The product includes technology for Data-flow analysis Unit test-case generation and execution, static analysis, regression testing, runtime error detection, code review, and design by contract. Jtest is used by companies such as Cisco Systems, TransCore, AIG United Guaranty and Wipro Technologies. It is also known to be used by Lockheed Martin for the F-35 Joint Strike Fighter program (JSF).[2]

D. Metasploit Project

Metasploit Project is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development. Its best-known sub-project is the open source Metasploit Framework, a tool for developing and executing exploit code against a remote target machine. Other important sub-projects include the Opcode Database, shellcode archive and related research. The Metasploit Project is well known for its anti-forensic and evasion tools, some of which are built into the Metasploit Framework.[7]

E. NMAP

Nmap (Network Mapper) is a security scanner, originally written by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich), used to discover hosts and services on a computer network, thus building a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host(s) and then analyzes the responses. The software provides a number of features for probing computer networks, including host discovery and service and operating-system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap can adapt to network conditions including latency and congestion during a scan. The Nmap user community continues to develop and refine the tool.Nmap started as a Linux-only utility, but porting to Windows, Solaris, HP-UX, BSD variants (including macOS), AmigaOS, and IRIX have followed. Linux is the most popular platform, followed closely by Windows. [2] [7]

F. Parasoft C/C++ Test

Parasoft C/C++test is an integrated set of tools for testing C and C++source code that software developers use to analyze, test, find defects, and measure the quality and security of their applications. It supports software development practices that are part of development testing, including static code analysis, dynamic code analysis, unit test case generation and execution, code coverage analysis, regression testing, runtime error detection, requirements traceability, and code review. It's a commercial tool that supports operation on Linux, Windows, and Solaris platforms as well as support for on-target embedded testing and cross compilers.[7] [3]

G. Soatest

SOAtest is a testing and analysis tool suite for testing and validating APIs and API-driven applications (e.g., cloud, mobile apps, SOA). Basic testing functionality include functional unit testing, integration testing, regression testing, system testing, security testing, simulation and mocking, runtime error detection, web UI testing, interoperability testing, WS-* compliance testing, and load testing.Supported technologies include Web services, REST, JSON, MQ, JMS, TIBCO, HTTP, XML, EDI, mainframes, and custom message formats.Parasoft SOAtest introduced Service virtualization via server emulation and stubs in 2002; by 2007, it provided an intelligent stubs platform that emulated the behavior of dependent services that were otherwise difficult to access or configure during development and testing. Extended service virtualization functionality is now in Parasoft Virtualize, while SOAtest provides intelligent stubbing.SOAtest is used by organizations such as Cisco, IBM, HP, Fidelity, Bloomberg, Vanguard, AT&T, IRS, CDC, Tata Consultancy Services, Comcast and Sabre.It was recognized as a leader in the Forrester Research's The Forrester Wave™: Modern Application Functional Test Automation Tools, Q4 2016, which evaluated 9 functional test automation tool vendors across 40 criteria. Forrester Research gave SOAtest the highest score among all vendors in the Current Offering category, citing its strength in API testing, UI automation, and key integrations. It also part of the solution recognized as "innovation and technology leader" in Voke's service virtualization market mover array.

V. SOFTWARE TESTING TECHNIQUES

There are eight software testing techniques. They are User Story Testing (AGILE), Use Case Testing, Checklist Based Testing, Exploratory Testing, Experienced Based Testing, User Journey Test, Risk-Based Testing, Heuristic Risk-Based Testing.

A. User Story Testing (Agile)

A user story can be described as a requested feature that is in the software from the perspective of the end user in agile software development life cycle. In user story, we have to specify the demand, the reason of the demand, and the user who is requesting it. Definition of Done (DOD) defines the completion criteria such as code is done, unit test is done, testing is done, UAT is done, etc… and Scrum guide states that Scrum team (developers, testers, PO etc.) owns and responsible for DOD. Also, acceptance criteria should be expressed clearly by POs (development team may help PO) and at least one test scenario for each acceptance criteria should be prepared in the test process of a user story and these acceptance criteria has to be tested carefully. The test entry and exit criteria must also be defined before starting the test run. [10]

B. Use Case Testing

A use case defines the operations that a user/actor performs in the system to achieve a specific purpose. Functional requirements of a system can be defined and managed using use cases. In this way, the scope of the desired or requested job is determined. Tests scenarios are prepared by taking into consideration the inputs and outputs of the steps determined by the user to reach a specific purpose. During the tests, the results of the tests are determined by comparing the expected outputs with the actual outputs. When writing Use-cases, generally business language is preferred instead of technical language. Therefore they are often used in writing acceptance tests. In order to cover all requirements**,** at least one test scenario is prepared for each requirement. By this way, test coverage can be increased and we can measure this coverage also by using traceability matrix. In the traceability matrix, we create a matrix table with test scenarios and requirements and put a cross sign in the relevant field if it meets the requirements for each test case. [10]

C. Chicklist Based Testing

In agile processes, we create ageneral checklist that is created independently from user stories. If there is not any risk is specified in the User Story, all or some of these checklist items may be used according to the scope of the user story. During execution of these tests, if you find a defect, you should extend the scope of the checklist by adding the failed scenario. Thus, we can increase risk items in the checklist for subsequent sprints. Example : All links in the system (Web / Mobile) should work correctly; There should not be agrammar error in the system writings; Font sizes, fonts, should be as expected; There should not be any pictures that cannot be loaded/broken in the system; Pictures, text, etc. The alignment between the other components must be as expected; All buttons must work properly and each of them directs the user to the corresponding operation; Each page should have the main page logo and should be redirected to the main page when clicked; Warning, information messages should be displayed in the correct format. If the page is responsive, it should be checked at all resolutions; All components on the site (dropdown, checkbox, radio button, etc.) should work correctly; Special conditions (numeric, alphanumeric, etc.) in the input fields must be checked; Operations cannot be performed by leaving the required fields blank; Any operation of the site must not last more than 3 to 15 seconds. [10]

D. Exploratory Testing

First of all, the exploratory testing is not a random or ad-hoc test**.** One of the biggest misconceptions about this test technique is that the exploratory testing is perceived as a random, non-testable, non-observable test technique. The exploratory testing is a test approach based on learning and exploring the product at the same time by using the experience, domain knowledge, analytical and intellectual knowledge of a test engineer in agile processes. Before starting exploratory testing, a preparation should be done**.** Regardless of exploratory testing method selection, we should prepare a plan for the scope of functionality, tools to be used, test data, environment etc. This plan will guide the tester during test execution process**.** Another important point of exploratory testing is documentation is fully completed after the tests are finished.Although it is not mandatory, “session-based testing” technique is generally preferred as an exploratory testing technique. This technique involves the following steps:Main Activities:Test Session Time (It should be a few hours); Session Activities: Session setup, Test Design and Test Execution, Defect Searching, Reporting; The purposes of the test should be specified; The goal of the test should be specified; Functionality included in the test (test report – charter) should be written. Test Report During & After the Test Process: Test Report (Charter) [Specifies the test function.], The Person who Performed Test, Start Date and Time, Session Metrics (Metrics Collected During Test Implementation and Defect Searching), Test Data, Test Notes, Results, Errors. [7] [10]

E. Experienced Based Testing

This testing technique based on the knowledge, skills, and experience of the person who will make the test. In this testing technique; test planning, test strategy, test inputs, and test scenarios are determined by the experience of the person performing the test. In order to prefer this technique, it must be an experienced candidate with sufficient technical and business knowledge to perform this test.It is easier to understand what is going right or wrong during the test because the experiences gained in past projects are taken into consideration. This person can perform tests by using techniques like exploratory testing, which will make it easier to use past experience and intellectual/analytical knowledge. When we have very short test execution time or there is a lack of sufficient documentation on the project, etc. make sense to use this test technique**.** If the system being tested comprises of high risks, it is not preferable to use the Experienced Based Testing technique alone because the context of the requirement should be totally covered. [11]

F. User Journey Test

The User Journey test is taking into account the various road maps and journeys of a typical user on the system. In these tests, the most critical journeys a user would make within the site are determined and then the scenarios of these journeys are written. Thus, the user’s interactions with the system are covered as much as possible**.** These tests are usually “end-to-end” tests, so they may take more time to run than other tests**,** but the coverage percentage of these tests are higherthan the other ones.Since the “User Journey” tests are comprehensive and extensive tests, the numbers are relatively low compared to other tests. Especially, “user journey” tests can be created taking into account the most critical “use-case” scenarios. The most plausible behavior here is that positive basic scenarios, called “happy-path”, should be run first.For an example on kariyer.net, it is animportant “user-journey” test for a user to enter the site, login, then search for the “sales representative” keyword and apply to the first job ad successfully. In this “user journey”, opening the site, logging in, making a successful call in the search bar, opening a related announcement page and then submitting the application from this announcement page is covered. As you can see here,the broad coverage of the “user journey” tests beneficial in early detection of critical faultsin the software development process especially before starting extensive testing. Unlike User Story tests, user journey tests are not tied to user stories. When there are changes introduced by a new user-story, existing user journey tests are updated taking these new changes into account**.** New user stories rarely cause new user journey tests. For this to happen, new features must be added to the system.

G. Risk- Based Testing

One of the most fundamental objectives of risk-based test techniques is to find the most critical and most important errors as early as possible with lowest cost. Risks are the things that we do not know exactly what will happen, but we know the probabilities of what might happen. Briefly, they are possible problems. When these possibilities are unknown, they are called uncertainties. Therefore, we can think of the general definition of the magnitude of the risk is the multiplication of the likelihood of problems and their impact. Thus, in risk-based tests, we prioritize and test the most error-prone functionalities. [Magnitude of Risk = Likelihood * Impact]. The early adoption of the risk-based test approach in projects is important for the early detection of critical problems.The most basic steps of the risk-based test are summarized below:First, risks are identified and a prioritized risk list is prepared; Make a test plan according to the prioritized risk list and tests are executed for each risk; As a result of the tests, some risks eliminate and some of them arise. New risks are tested by considering their test effort**.**If you are responsible for testing a product with a very high failure cost, you will need to do a very detailed risk analysis. Statistical models can be used here. One of the most known model is the Failure Mode Effect Analysis (FMEA) model. [10] [6]

H. Heuristic Risk-Based Testing

At the beginning of your projects, your risk analysis may be incomplete or incorrect in some degree because it is not possible to estimate everything 100% at first. However, as your project progresses and your product improves, your estimates and risk analysis will become increasingly stronger. According to James Bach, the two most critical factors for risk are experiences and teamwork. Over a period of time, products or technologies begin to reveal characteristic problems. It is important to observe and learn about them. Besides, it is very critical to do risk analysis with people with different perspectives. Also, using risk-based lists helps us to negotiate with the management about test workforce effectiveness. We can show them that we use existing test workforce for the most critical points of the product or we may need to explain them we need the extra workforce to cover all the risk areas. These risk-based lists give you extra power to negotiate with the management.In this article, we covered very important test techniques. I hope you enjoyed to read it. All the things in this article based on my understanding and experiences. If you disagree anything, please do not hesitate to write a comment and if you convince me I will fix the thing that you are right.

VI. ADVANCED SOFTWARE TESTING TECHNIQUES

Software testing is a vital process in the software development life cycle, but it tends to become a hectic process on when performed a daily basis.To help me out, I’m going to share with me five advanced software testing techniques that will help I make my day more efficient and productive. Together, we’ll cover systematic and some non-systematic software testing techniques.Want to know how exactly these testing techniques work? Then keep reading to find out. [9] [2]

A. Identification of Test Scenarios

The basis of our entire testing process is dependent on our testscenarios. QA team should sit together and come up with possible test scenarios. This would maximize the chances of covering majority test cases. To make sure that our test casescover the maximum functionality of the system, use the following tips: Go through the specifications documentmeticulously and derive test cases; If there are conditions applied, create a test case that makes every condition once true and once false; If there are several branches of any process, make sure that test case cover every branch at least once; Make sure we know the ‘Expected Result’ for each case; Same action cannot have two outputs; Identify scenarios for cross-functional testing. [3] [9]

B. Use Case-Based Testing

We can also get through our day more efficiently by using use case-based testing. In this technique, test cases are developed using the use cases of the system. A use case encompass the various actors and their interactions with the system. Use cases cover the complete transactions from start to finish. These test cases depict the actual use of software by the end user. Thesetest cases are significant for the acceptance of software by the users.Our software testing process will be expedited if you already have use cases documented for your project. To apply use case based technique, follow these steps:Develop at least one test case for a normal use; Cover all use cases: Cover all users; Identify any dependency between different use cases and cover those; Develop a test cases for scenarios that have not been identified in the use case. [9]

C. Decision Table Testing/ Cause Effect Graph

A decision table is also referred as a Cause-Effect table. This technique is used for the functions which respond to a combination of inputs or events. For example, a Submit button on form will only be enabled if the user has entered all required fields. In order to use decision table testing and to make your software testing effective, the first task is to identify such functionalities where output depends on a combination of inputs.If there is large input set of combinations, divide it in smaller subsets so that the decision table remains manageable. For each function, create a table and list down all possible combinations of inputs and respective outputs.This might also help in identifying any condition that is overlooked by the requirements analyst. Follow below steps to create a decision table: Enlist the inputs in rows; Enter the rules in the column; Fill the column with different; combination of input; In the last row, note down the output against the input combination.[2] [9]

D. Automated Testing

Automated Testing is becoming popular to make the testing process efficiently. It unloads the testing effort from human testers and assures that no test case has been missed.We can utilize this software testing technique by keeping the following points in mind:Automate the routine tasks; Automate validations for mandatory fields; Automate date field validations, if there are any; Automate input type and input length validations; Automate the process of checking dead links; Automate your decision tables. [3] [4] [9]

E. Experience- Based Testing

Nothing beats the human mind armed with the experience. Make your day more efficient by combining experience with the talent. If you have worked on any other similar project before, use the experience gained by that project into use. Experience-based software testing technique can be used when the experienced resources share their knowledge with the new and relatively inexperienced testers. Experienced testers can help the team in the following ways: A fresh tester may execute a test case once. But an experienced tester knows thathe may create memory issues by repeating the same action several times; A fresh tester may execute a test case and passes a functionality. But an experienced tester knows that if he enters a particular input that could break the functionality; From experience, you know that it happens that same functionality is implemented in an un-identical manner throughout the application. So, it would be required to check the same functionality at every place.Experience based testing enables one to make error guesses. You can look at an error and anticipate the possible faults in the system. The next steps would be to define the test cases designed specifically to expose those errors. [2] [3] [9]

CONCLUSION

This paper on Software testing describes in detail about software testing, need of software testing, Software testing goals and principles. Software testing is often less formal and rigorous than it should, and a main reason for that is because Iam studying to define best practices, methodologies, principles, standards for optimal software testing. To perform testing effectively and efficiently, everyone involved with testing should be familiar with basic software testing goals, principles, limitations and concepts. Static testing tools and dynamic testing tools are impact tools.

References

[1] https://www.guru99.com/Testing

[2] William Jackson, published by 2009/02/09; https://gcn.com/articles/static-vs-dynamic-code-analysis.aspx

[3] http://www.softwaretestingclass.com/software-testing-tools-list/

[4] https://www.upwork.com/Testing

[5] http://istqbexamcertification.com/what-is-software-testing/

[6] Dinesh Thakur, Category: Software Engineering/http://ecomputernotes.com/software-engineering/software-testing-tools

[7] https://en.wikipedia.org/wiki/Category:Software_testin g tools,From Wikipedia, the free encyclopedia

[8] https://en.wikipedia.org/wiki/Category:Security testingtools, From Wikipedia, the free encyclopedia

[9] Ulf Eriksson, 27th May 2016 Testing,https://reqtest.com/testing-blog/advanced-software-testing-techniques/

[10] Onur Baskirt,Published by March 30th, 2017, Head of Software Testing and Manager of two Development Teams at Kariyer.net, http://www.swtestacademy.com/software-testing-techniques/

[11] James Bach, First published in Software Testing and Quality Engineering Magazine, 11/99 Copyright 1999, http://www.satisfice.com/articles/hrbt.pdf

[12] Lu Luo, Institute for Software Research InternationalCarnegie Mellon UniversityPittsburgh, PA15232, USA

[13] Jan Tretmans, Nijmegen Institute for Computing and Information Sciences, Radboud University Nijmegen The Netherlands1Version.

[14] Abhijit A. Sawant, Pranit H. Bari, P. M. Chawan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.980-986 986

[15] Vishal D Rampure, Term Paper on Testing tools, Advanced Software Engineering CSC

[16] Jack,Birmingham, http://istqbexamcertification.com/what-is-static-analysis-tools-in-software-testing/

[17] Lakshay Sharma, published byMay 16, 2016; http://toolsqa.com/software-testing/static-testing

[18] Sakshi Dewan, by STC team member;Difference between static testing and dynamic testing**.**