International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 04, 2020
ISSN: 1475-7192

A Study on Risk Management in Software Testing

[1]Dr Leelavathi Rajamanickam, [2]Dr Norriza Hussin, [3]Kate Lam Woon Yee

***Abstract*—** *Software testing is accounted to be an essential part in software development life cycle in terms of cost and manpower, where its total cost is considerable high. Consequently, many studies have been conducted to minimize the risk associated cost and human effort to fix bugs and errors, and to improve the quality of testing process. Risk can be internal risk and external risk. Risk management has to identify the risk, reduce the impact of risk and monitoring of risk. Test cases can be generated from different phases (requirement phase, design phase and after development phase). Though, test case generation at early stages is more effective rather than that after development, where time and effort used for finding and fixing errors and bugs is less than that after development. At later stage, fixing errors results in enormous code correction, consuming a lot of time and effort. In this paper, we study the different paradigms of testing techniques for generating test cases, where we investigate their coverage and associated capabilities. We finally propose a preliminary model for a generic automated test cases generation.*

***Keywords*—** *Software testing; risk management; testing principles; testing limitations*

I. INTRODUCTION

Software testing is about testing a feature with varying test data to get a result and then comparing the actual result with expected result, it is not merely finding defects or bugs in the software; it is the completely dedicated discipline of evaluating the quality of the software (Leelavathi Rajamanickam., 2014). Software testing is a phase of SDLC that entails much effort, time and cost. Often, testing phase is the single largest contributor towards the whole development time. Testing can not only uncover bugs in the program, but also flaws in design of the software (Leelavathi Rajamanickam., 2014). Software testing is used to ensure high quality software. This is done to detect defects in the SUT that can cause software failures. The process is time-consuming and complex. It can consume about 50% of the total cost. Software testing can be defined as the process of validating and validating SUT to meet technical

[1] School of IT, SEGi University, Malaysia
[2] School of IT, SEGi University, Malaysia
[3] School of IT, SEGi University, Malaysia

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

and business expectations. Software testing has evolved over the decades. The growing maturity of applications and underlying technologies forces the test community to continue to optimize. Early developers were responsible for testing, and because the application required higher reliability, independent test teams were established to ensure that reliability was a key criterion for all phases of the project. Over time,

people realized that an independent test team could provide a different perspective and help identify more defects. Today, business scenarios require greater flexibility and require us to recognize financial cuts while ensuring quality. Testing began as manual testing in the 19th century and evolved in the 21st century with automated testing of hybrid/keyword frameworks.

In this day and age, software testing in software projects typically costs 30-50% of the project budget. As software complexity increases, testing becomes very expensive, leading to high risk, because a poor software testing process always leads to failures, and software testing in software projects typically costs 30-50% of the project budget. As software complexity increases, testing becomes very expensive, leading to high risk, because a poor software testing process always leads to frequent system downtime, higher maintenance costs, rework, and even serious lawsuits. With the increasing complexity of modern software projects, traditional manual test suites have become unwieldy and brittle. With the increasing complexity of modern software projects, traditional manual test suites have become unwieldy and brittle. Software validation is used to check that the SUT is compliant and similar to the structure test. Validation is done by running SUT, which is similar to functional testing. Generally speaking, there are four main methods of software testing: black box testing, white box testing, grey testing, and functional testing. Software tests can be performed manually or automatically using test tools. Automated software testing has been found to be better than manual testing. Recently, more and more test case generation tools have been available. These tools use a variety of methods to perform their tasks.

II. LITERATURE REVIEW

Software testing is really required to point out the defects and errors that were made during the development phases. This is necessary because it ensures customer reliability and their satisfaction in the application. And also, it is important to guarantee the quality of the product, providing quality products to customers helps gain their trust. 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. It is important to ensure that the application does not cause any failures, as this can be very expensive in the future or in the later stages of development. Last but not least, it’s important because it is required to stay in the business.

III. RISK MANAGEMENT IN SOFTWARE TESTING

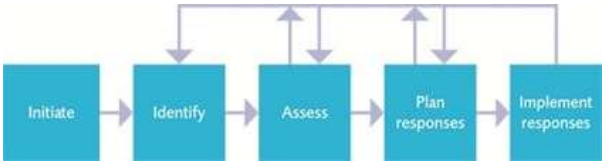
Imagine working on a project for over four hundred hours, and then a small bug destroys the entire software, that could've been prevented if taken care of within 1 minute. The only option available in this situation is to remove the product from the market completely and start fresh. To manage the risks we need to establish a strong bond between the

customers and the team members. A risk management strategy can be included in the software project plan or the risk management steps can be organized into a separate Risk Mitigation, Monitoring and Management Plan. Software metrics and tools can be developed to manage the risks.

Q*uality research* is very important. Nowadays, the software development is extremely competitive, making the customer expect the best out of every application they come across or they could delete it and search for a similar one. Software should be able to adapt to so many platforms and operating systems including mobiles, laptops, windows, IOS, etc. The real challenge with software testing is time. Developers usually don't usually have enough time to develop high quality projects and quality testing at the same time. So, the only safe solution is to limit tests within specific times and resources. So how can testers know which the important aspects are? What are the risky components? From whose perspective is the importance of the feature or functionality decided? To answer all these questions, an "RBT" should be formed. Risk can be viewed as an opportunity to develop our projects. The project should be managed in such a way that the risks don't affect the project.

IV. RISK CATEGORIZATION

Risk Management Process follows the five basic steps.



Identify the Risk. You and your team uncover, recognize and describe risks that might affect your project or its outcomes. There are a number of techniques you can use to find project risks. During this step you start to prepare your Project Risk Register.

Analyze the risk. Once risks are identified you determine the likelihood and consequence of each risk. You develop an understanding of the nature of the risk and its potential to affect project goals and objectives. This information is also input to your Project Risk Register.

Evaluate or Rank the Risk. You evaluate or rank the risk by determining the risk magnitude, which is the combination of likelihood and consequence. You make decisions about whether the risk is acceptable or whether it is serious enough to warrant treatment. These risk rankings are also added to your Project Risk Register.

Treat the Risk. This is also referred to as Risk Response Planning. During this step you assess your highest ranked risks and set out a plan to treat or modify these risks to achieve acceptable risk levels. How can you minimize the probability of the negative risks as well as enhancing the opportunities? You create risk mitigation strategies, preventive plans and contingency plans in this step. And you add the risk treatment measures for the highest ranking or most serious risks to your Project Risk Register.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

Monitor and Review the risk. This is the step where you take your Project Risk Register and use it to monitor, track and review risks.

Risk is about uncertainty. If you put a framework around that uncertainty, then you effectively de-risk your project. And that means you can move much more confidently to achieve your project goals. By identifying and managing a comprehensive list of project risks, unpleasant surprises and barriers can be reduced and golden opportunities discovered. The risk management process also helps to resolve problems when they occur, because those problems have been envisaged, and plans to treat them have already been developed and agreed. You avoid impulsive reactions and going into "fire-fighting" mode to rectify problems that could have been anticipated. This makes for happier, less stressed projec

V. RISK MANAGEMENT STANDARDS

A number of standards have been developed worldwide to help organisations implement risk management systematically and effectively. These standards seek to establish a common view on frameworks, processes and practice, and are generally set by recognised international standards bodies or by industry groups. Risk management is a fast-moving discipline and standards are regularly supplemented and updated.

The different standards reflect the different motivations and technical focus of their developers, and are appropriate for different organisations and situations. Standards are normally voluntary, although adherence to a standard may be required by regulators or by contract.

IRM professional qualifications seek to equip students with the knowledge and judgement to select the appropriate standard or standards for use within their organisation.

Commonly used standards include:

- ISO 31000 2009 – Risk Management Principles and Guidelines
- A Risk Management Standard – IRM/Alarm/AIRMIC 2002 – developed in 2002 by the UK's 3 main risk organisations.
- ISO/IEC 31010:2009 - Risk Management - Risk Assessment Techniques
- COSO 2004 - Enterprise Risk Management - Integrated Framework
- OCEG "Red Book" 2.0: 2009 - a Governance, Risk and Compliance Capability Model

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

VI. RISK BASED TESTING

Risk Based Testing is to prevent the scenarios that could impact the business negatively through prioritizing main testing in coding functions and executing designs. RBT should be applied early in the software developing life cycle, to ensure that major bugs will not occur with the end user. If this technique was not used, customers could get very frustrated or even lose interest in the business. Minor and mild bugs could be annoying for the customer, they can write a complaint about their error and developers will fix it as soon as possible, but when major errors occur, customers will delete the software and will give a bad review to the product, which will give the software a bad reputation. To avoid all that, emphasizing on RBT while going through all the phases will help provide a foundation of the software with minimal major fails.

RBT is when the team identifies the real major risks that could be the most destructive to the software, and give it to the testers. This process will reduce the time wastage for testers by a lot. Obviously, RBT should be approached when there is a limitation or constraint on time, cost, and resources within a project. RBT is useful when the program gets complexed and challenging

REFERENCES

1. Leelavathi Rajamanickam, "Testing Tool for Object Oriented Software" International Journal of scientific research and management (IJSRM), vol 2. Issue 8 august 2014, pages: 1205-1208 (2014).
2. Leelavathi Rajamanickam, “Software Testing, Analysis and Objectives” International Journal of Advanced Trends in Computer Science and Engineering, **Vol 3**, No.5, Pages: 01-04 (2014).
3. Ghahrai, A. (2018). *Seven Principles of Software Testing — Testing Excellence*. [online] Testing Excellence.
4. Awan, k., Shah, J., Akram, A., Gupta, n., Developer, P., Kim, H. and Adke, R. (2018). Types of Software Testing: DifferentTesting Black, R. and Graham, D. (2018).
5. Bertolino, A. (2018). *Software testing research and practice*. [online] Dl.acm.org. Available at: https://dl.acm.org/citation.cfm?id=1754751 .
6. Myers, Glenford J., —” The art of software testing”, NewYork: Wiley, c1979. ISBN: 0471043281.
7. Peter Sestoft, “Systematic software testing”, Version 2,
8. 2008-02-25.
9. Rajat Kumar Bal, “Static and Dynamic Testing Compared”., “Software Testing”.
10. Shari Lawrence Pfleeger, —” Software Engineering, Theory and Practice”, Pearson Education, 2001.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

11. COHEN, M. B., SNYDER, J., AND ROTHERMEL, G. 2006. Testing across configurations: Implications for combinatorial testing. SIGSOFT Softw. Engin. Notes 31, 6, 1–9.

12. Kuhn, D.R.; Kacker, R.; Lei, Y. Practical Combinatorial Testing, NIST SP800-142, National Institute of Standards and Technology, October 6, 2010.

13. Zamli, K.Z., Othman, R.R., Zabil, M.H.M., On Sequence Based Interaction Testing, IEEE Symp. on Computers and Informatics, IEEE, 20-23 Mar. 2011, pp. 662-667.