International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Risk Assessment Model and its Integration into an Established Test Process

Ashwani Kumar[1], Prince Sood[2]

[1]M.Tech Student, [2]Assistant Professor

[1,2]Computer Science and Engineering, Universal Institution of Engineering & Technology, Lalru, Punjab, India

How to cite this paper: Ashwani Kumar | Prince Sood "Risk Assessment Model and its Integration into an Established Test Process" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.1507-1512, **https://doi.org/10.31142/ijtsrd26757**

IJTSRD26757

ABSTRACT

In industry, testing has to be performed under severe pressure due to limited resources. Risk-based testing which uses risks to guide the test process is applied to allocate resources and to reduce product risks. Risk assessment, i.e., risk identification, analysis and evaluation, determines the significance of the risk values assigned to tests and therefore the quality of the overall risk-based test process. In this paper we provide a risk assessment model and its integration into an established test process. This framework is derived on the basis of best practices extracted from published risk-based testing approaches and applied to an industrial test process.

KEYWORDS: *Risk Assessment; Risk Identification; Risk Analysis; Risk Evaluation; Risk-Based Testing; Risk Management; Software Testing.*

INTRODUCTION

Test case prioritization organizes check cases in an exceedingly} very because of accomplish some performance goals with efficiency. Rate of fault detection is one amongst most important performance objective. The check cases ought to run in degree order that raises the possibility of fault detection among the sooner life cycle of testing. Action prioritization techniques have tested to be useful in rising regression testing activities. throughout this paper, we've projected degree algorithmic rule, that prioritizes the system check cases supported the six factors: shopper priority, variations in demand, implementation issue, demand bit ability, and execution time and fault impact of demand. We tend to tend to conducted a controlled experiment on two industrial data sets to ascertain the projected worth based action Prioritization algorithmic rule with random prioritization for early rate of fault detection. Average share of fault detection metrics has been accustomed live the efficiency of projected and random prioritization and it shows that the projected worth based algorithmic rule is further economical than random prioritization to come back up with sequence of check cases for early rate of fault detection. Presently a day's code enlargement setting testing has come back enclosed as a result of shorter product Time to market, shrinking budgets and better-quality demands. Code checking could be a modern time the flexibility level of take a look at personnel and size of the project impact intense methodology, that's sometimes restricted by worth and time constraints, defect detection rate. As can testability of the system.

This is as a result of poorly mere wants with inadequate description of user feedback. this may be considerably apparent under control automation where take a look at cases have religion in user feedback thus on execute after all. code testing drives are normally beleaguered by constraints like time, cost, and deficient Skills. These constraints impose risk on the idea of software system check effectiveness with connexion software package testing aims. Sympathetic the thanks to moderate this risk may well be a key-factor in achieving victorious software system testing. Our exploration aims to find software system check tools.

Numerous software development and testing methodologies, tools, and techniques have emerged over the previous few decades promising to bolster software quality. Software testing may be a trade-of between budget, time and quality. However, as a result of the observe of software development has evolved; there has been increasing interest in increasing the role of testing upwards inside the SDLC stages, embedding testing throughout his systems development technique. The speedy modification inside the software Development technique brings many challenges to the present field. In instruction to return back across these challenges, the companies sought for extra agile and price effective ways in which. This angle is visible altogether phases of software development technique. The ways, approaches and techniques of software testing have developed to adapt his modification.

Background on Risk-Based Testing

Testing is a field, which is well researched and surveyed, relatively, within software engineering. We study 12 research papers for our research and surveyed 8 organizations in their review. We identified empirical studies, evaluating from 8 organizations 5 organizations favored risk-based test cases selection rest of 3 organizations favored design-based test cases selection. Less than a third of the studies comprise Industry scale contexts.

Since the area is well reviewed recently, we here only focus on the work closely related to the topic under study, namely, empirical evaluations in industry on regression test prioritization and selection.

Concept of Risk

A risk is that the likelihood of injury, harm or loss and usually determined by the chance of its incidence and its impact. because it is that the likelihood of one thing happening that may have a sway on objectives , the quality risk rationalization is predicated on the 2 factors chance (P), determinant the chance that a failure assigned to a risk happens, and impact (I), determinant the value or severity of a failure if it happens operational. Mathematically, the chance exposure R of associate capricious quality a, i.e., one thing to that a celebration assigns worth, is decided supported the chance P and also the impact I within the following way:

$R(a) = P(a) \circ I(a)$

In the context of testing, assets ar capricious testable artifacts conjointly referred to as risk things. for example, necessities, components, security risks or failures ar typical risk things to that risk exposure values R further as tests ar assigned . at intervals testing, a risk item is assigned to check cases that ar usually related to risk exposure values themselves derived from the chance items' risk exposure values. Risk exposure is typically conjointly referred to as risk coefficient, risk worth or not distinguished from the chance itself. The represented operation ◦ represents a multiplication of 2 numbers or a vector product of 2 numbers or letters (and will chiefly be associate capricious computing accustomed confirm risk). The factors P and that i is also determined directly via appropriate metrics or indirectly via intermediate criteri supported the Factor-Criteria-Metric model . The chance usually considers technical criteria like quality of parts assigned to the chance item and also the impact considers business criteria like financial loss. The metrics are often measured mechanically, semi-automatically or manually. for example, the quality of a element are often calculable mechanically by the McCabe quality and also the financial loss are often calculable manually by a client. supported the determined metrics, risk exposure values ar computed on the premise of a calculation procedure. Finally, risk exposure values ar assigned to risk levels. A risk level indicates the criticality of risk things and serves the aim to match risk things further on confirm the employment of resources, e.g., for testing. Risk levels ar usually denned via risk matrices combining chance and impact of a risk. Associate example for a risk matrix is shown in Fig. 1.

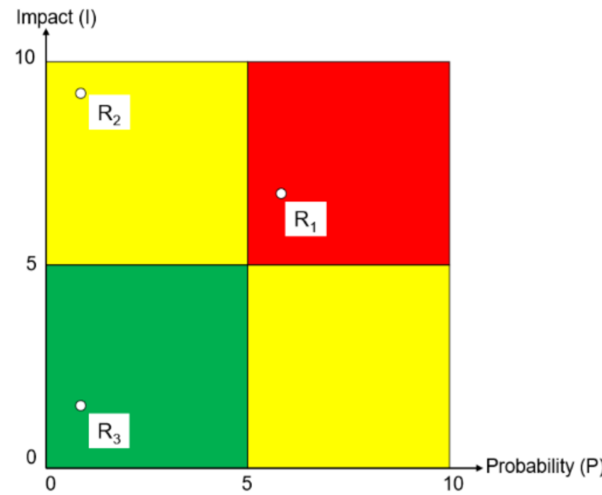

Fig 1.1 Risk Matrix Examples

The 2x2 risk matrix of Fig. 1. Probability and impact range from 0 to 10 and are shown on the x-axis and y-axis, respectively. Items in the lower left cell ([0..5]×[0..5]) have low risk, items in the upper right cell ([5..10]×[5..10]) have high risk, and items in the remaining cells ([0..5] × [5..10] and [5..10] × [0..5]) have medium risk. For instance, risk R1 in Figure 1 with value 6×7 is high, R2 with value 1×9 is medium, and R3 with value 1×2 is low.

Risk-Based Testing Approaches

The overall purpose of RBT approaches is to check in Associate in nursing efficient and effective means driven by risks. As mentioned before, each offered risk-based testing approach thus integrates testing and risk assessment activities. Many RBT approaches are planned in scientific conferences and journals. we tend to consistently extracted these approaches from comprehensive connected work sections of 4 recently printed journal articles on risk-based testing to urge a broad and representative summary of RBT approaches. we tend to thought-about all RBT approaches denned within the journal articles themselves moreover as all RBT approaches cited in a minimum of one connected work section of the four journal articles. to ensure proof of the approaches and enough details to extract relevant info, we tend to thought-about solely RBT approaches reportable in papers with a length of a minimum of four pages printed in a very sciatic journal or in conference proceedings. Table one lists all collected RBT approaches ordered by the date of their first publication. Some approaches, i.e., Redmill, Stall Baum, Souza, moreover as Felderer and Ramler square measure lined by quite one cited publication (see entries with identifiers 03, 04, 05 and thirteen in Table 1). Most listed approaches square measure cited by quite one journal article that is a further indicator for the connectedness of the RBT approaches collected in Table 1.

Table 1: Overview of Identified Risk-based Testing Approaches

Id	Approach	Description
1	Amland	The approach defines a process which consists of the steps (1) planning, (2) identification of risk indicators, (3) identification of cost of a fault, (4) identification of critical elements, (5) test execution as well as (6) estimation to complete. In addition, it is presented how the approach was carried out in a large project
2	Chen et al.	The approach defines a specification-based regression test selection with risk analysis. Each test case is a path through an activity diagram (its elements represent requirements attributes) and has an assigned cost and severity probability. The test selection consists of the steps (1) assessment of the cost, (2) derivation of severity probability, and (3) calculation of risk exposure for each test case as well as (4) selection of safety tests. The risk exposure of test cases grouped to scenarios is summed up until one runs out of time and resources. The approach is evaluated by comparing it to manual regression testing
3	Redmill	The approach reflects on the role of risk for testing in general and proposes two types of risk analysis, i.e., single-factor analysis based on impact or probability as well as two-factor analysis based on both factors.
4	Stallbaum et al.	The approach is model-based. Risk is measured on the basis of the Factor-Criteria-Metrics model and annotated to UML use case and activity diagrams from which test cases are derived.
5	Souza et al.	The approach defines a risk-based test process including the activities (1) risk identification, (2) risk analysis, (3) test planning, (4) test design, (5) test execution, as well as (6) test evaluation and risk control. In addition, metrics to measure and control RBT activities are given. The approach is evaluated in a case study.
6	Zimmermann et al.	The approach is model-based and statistical using Markov chains to describe stimulation and usage profile. Test cases are then generated automatically taking the criticality of transitions into account. The approach focuses on safety-critical systems and its application is illustrated by examples.
7	Kloos et al.	The approach is model-based. It uses Fault Tree Analysis during the construction of test models represented as state machine, such that test cases can be derived, selected and prioritized according to the severity of the identified risks and the basic events that cause it. The focus of the approach is safety-critical systems and its application is illustrated by an example.
8	Yoon and Choi	The approach defines a test case prioritization strategy for sequencing test cases. Each test case is prioritized on the basis of the product of risk exposure value manually determined by domain experts and the correlation between test cases and risks determined by mutation analysis. The effectiveness is shown by comparing the number and severity of faults detected to the approach of Chen et al.
9	Zech	The approach is model-based and derives a risk model from a system model and a vulnerability knowledge base. On this basis a misuse case model is derived and test code generated from this model is executed. The approach is intended to be applied for testing cloud systems.
10	Bai et al.	The approach addresses risk-based testing of service-based systems taking the service semantics which is expressed by OWL ontology into account. For estimating probability and impact dependencies in the ontology are considered. The approach considers the continuous adjustment of software and test case measurement as well as of rules for test case selection, prioritization and service evaluation. The approach is evaluated by comparing its cost and efficiency to random testing.
11	Felderer et al.	The approach defines a generic risk-based test process containing the steps (1) risk identification, (2) test planning, (3) risk analysis, (4) test design as well as (5) evaluation. Steps (2) and (3) can be executed in parallel. For this test process a risk assessment model based on the Factor-Criteria-Metrics model is defined. The metrics in this model can be determined automatically, semi-automatically or manually. The approach is illustrated by an example.
12	Wendland et al.	The approach is model-based. It formalizes requirements as integrated behavior trees and augments the integrated behavior tree with risk information. Then for each risk an appropriate test directive is identified, and finally both the risk-augmented integrated behavior tree and the test directive definition are passed into a test generator.
13	Felderer and Ramler	The approach defines a process to stepwise introducing risk-based testing into an established test process. On this basis four stages of risk-based test integration are defined, i.e., (1) initial risk-based testing including design and execution of test cases on the basis of a risk assessment, (2) risk-based test results evaluation, (3) risk-based test planning, as well as (4) optimization of risk-based testing. The approach is evaluated in a case study.
14	Ray and Mohapatra	The approach defines a risk analysis procedure to guide testing. It is based on sequence diagrams and state machines. First one estimates the risk for various states of a component within a scenario and then, the risk for the whole scenario is estimated. The key data needed for risk assessment are complexity and severity. For estimating complexity inter-component state-dependence graphs are introduced. The severity for a component within a scenario is decided based on three hazard techniques: Functional Failure Analysis, Software Failure Mode and Effect Analysis and Software Fault Tree Analysis. The efficiency of the approach is evaluated compared to another risk analysis approach.

Risk Assessment Framework

In this section we present a risk assessment framework for risk-based testing purposes. This framework is shown in Fig. 3. It contains a risk assessment model which configures the risk-based test process. The execution of the test process provides feedback to continuously refine and improve the risk assessment model. As mentioned in the previous section, the risk-based test process integrates risk assessment into the test process and uses risks to support all phases of the test process, i.e., test planning, design, implementation, execution, and evaluation. The framework is based on the risk-based test process which is configured by and provides feedback for the risk assessment model and explained as background in Section 2.

The risk assessment model and its elements therefore determine the overall risk-based test process and are the main component of our risk assessment frame- work for testing purposes. The risk assessment model defines the test scope, the risk identification method, a risk model and the tooling for risk assessment. In the following, we explain these elements in more detail illustrated by examples from the RBT approaches collected in Section 2.3. Each mentioned approach is referred to by its name and identifier. For the often-cited approach of Amland [6] we discuss all aspects of risk assessment model definition.

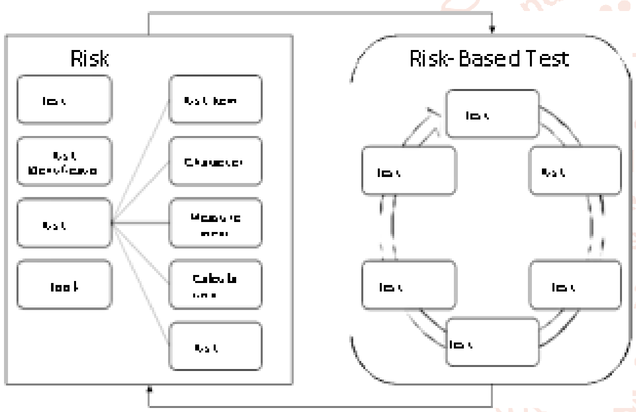

Fig.1. Risk Assessment Framework

PROPOSED WORK

Software testing could be a method during which we have a tendency to make sure that developed computer code is error free and playing of course. during this method, we have a tendency to produce take a look at cases that have some predefined results. we have a tendency to use these take a look at cases {to take a look at to check} computer code by comparison predefined results with the results obtaining by running these test cases. Computer code testing could be a important section of the computer code development life cycle. Quite seventieth time of the computer code development life cycle took this section. The computer code will not hundred % bug-free however playing computer code take a look at can cut back error kind the computer code. Risk-based testing could be a form of testing during which functions of the computer code area unit tested supported the priority, importance, and chance of the prevalence of a selected risk. During this approach, to check a computer code list of risk is ready alongside the varied risk parameters (the priority, importance, and chance of the occurrence). Supported these risk parameters risk level or risk issue is calculated that is employed in type listing the take a look at cases.

PROPOSED METHODOLOGY

The implementation of the proposed system is done in the JAVA to develop a risk-based test system. The main features of the proposed system are:

- Give more attention to the risks of the project instead of the functionality of the project.
- Help in estimating the time required for a particular project.
- Allow project manager to calculate the total cost of the project.
- Reduce the number of a test case by selecting only those test cases which have risk factor more than the threshold value (threshold value defined by tester).
- Help in estimating how much a project can delay if a particular risk will occur in the system.
- Help testing team and improve customer satisfaction.
- Improve the quality of all critical functions of the applications are tested.
- Help in creating test coverage. By using this test can know what has/has not been tested.

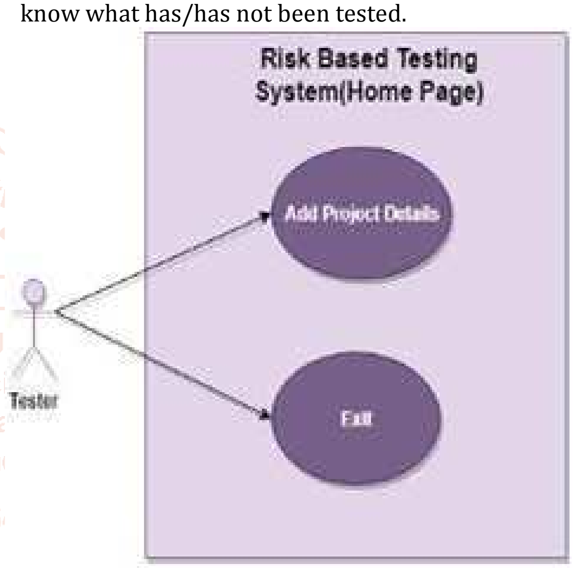

Fig. 2: Use case of Home page for developed application

Figure 2 is representing the Use-Case diagram for the home page of the Risk-Based Testing System. The home page of the system has only two components with which the tester will interact and that are about adding the project details and the second component is for quitting the home page.

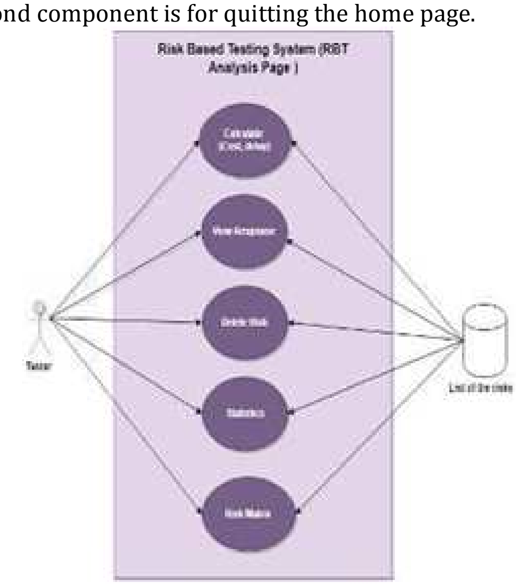

Figure 3: Use case of Risk-Based Testing application

Figure 3 is representing the Use-Case diagram for the analysis page of the Testing Application. There are many components with which the tester will interact like calculation of the time and space trade-off, statistics, risk identification, and risk matrix etc. The data for the analysis of the risks will be taken from the database which is named as the 'list of the risks.'

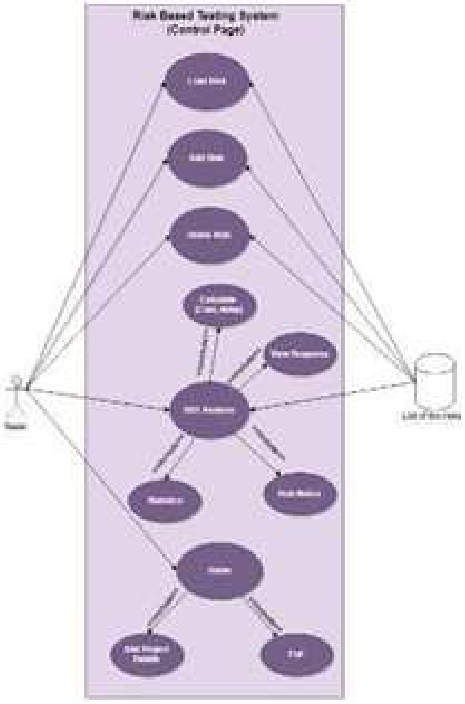

Figure 4: Use case for the control page of the application

Figure 4 represents the Use-Case Diagram for the Control Page of the application. This diagram further provides the exploration of the analysis page. This diagram is showing the complete overview of the risk-based testing application which also includes the involvement of the database.

RESULT & DISCUSSION

The results of the developed application are shown below. It includes the assorted choices that are provided for testing the device software package and gathering the desired results so correct call will be created in time while not compromising the standard and time of the project.

Figure 5: Main Page for Application

This Figure 5 shows the main screen of the application. As it can be clearly seen, this interface includes the two buttons i.e. Add Project Details and the Exit button. On the action for the Add Project Details button, the screen shown in the next figure will appear.

Figure 6: Control Page of Application

In the next screen as shown in Figure half-dozen, the small print of the project like project name, value and time are displayed. There square measure controls which is able to facilitate within the analysis of the risks, as shown within the Figure like Load risks, Add Risk, Delete Risk and also the RBT Analysis. because the tester can within the next Figure 7.

Figure 7: Risk Identification Screen

On the click on the Load Risks, this screen will appear, in which the information about the risk will be added like Risk ID, Cause for the Risk, and Response for the Risk. The probability will also be added which will let the tester know the tendency of the risk that can happen and its impact on the project will also be determined.

RISK MATRIX

Figure 11: Graph for obtained results in risk analysis

The graph in Figure 11 has shown the complete result of the analysis of the risk in the Risk-Based Testing. The bars in the graph will show the impact of the risk on the cost and the deadline for the risks associated with the particular project. This will also show the probability of the risk that can affect the progress of the project.

CONCLUSION & FUTURE WORK

The risk-driven approaches play a serious role within the testing of device code. there's a requirement to be a lot of aware concerning the danger that's related to the project in order that during a later cycle of the code development, it mustn't have an effect on the price of the project significantly. With this objective, the analyzation of the danger has become necessary within the code life cycle. This paper presents a replacement approach to check the device code victimisation the risk-based testing. The planned code in JAVA language is ready to search out the impact of the danger on the device code and additionally suggests the attainable alternatives which will be taken to avoid or scale back that risk. The approach can facilitate the testers to check the code supported varied pre-defined risks and therefore the user may also enter new risks within the system still. within the future, the code application is extended more give to supply to produce} additional choices check to check} the device code prefer to provide choices to manage code in parallel and to supply additional choices to manage and test the risks.

REFRENCES

[1] N. Trapani, M. Macchi, and L. Fumagalli, "Risk driven engineering of Prognostics and Health Management systems in manufacturing," IFAC-PapersOnLine, vol. 48, no. 3, pp. 995–1000, 2015.

[2] P. Kębłowski and A. Welfe, "A risk-driven approach to exchange rate modelling," Economic Modelling, vol. 29, no. 4, pp. 1473–1482, 2012.

[3] R. Kolb, "A risk driven approach for efficiently testing software product lines," in: 5th GPCE Young, Researches Workshop, Erfurt, Germany, and September 2003.

[4] G. Fairbanks, "A risk driven model for agile software architecture," Rhino Research, pp. 1–11, 2010.

[5] S. Laeeq and R. Shahid, "A Survey on Risk-Based or Design-Based Test Cases Prioritization ," International Journal of Scientific & Engineering Research, vol. 5, no. 12, pp. 400–403, Dec. 2014.

[6] M. F. Wendland, I. K. Schleferdecker, and M. Kranz, "A systematic approach to risk based testing using risk annotated requirements model," ICSEA 2012, pp. 636–642, Jan. 2012.

[7] A. Meriem and M. Abdelaziz, "A Meta-Model for Risk-Based Testing Technique Based On the MDA Approach," International Journal of Scientific & Engineering Research, vol. 9, no. 1, pp. 1–11, 2018.

[8] M. Felderer, C. Haisjackl, V. Pekar, and R. Breu, "A Risk Assessment Framework for Software Testing," Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications Lecture Notes in Computer Science, pp. 292–308, 2014.

[9] M. Lehto, "The concept of risk-based testing and its advantages and disadvantages," IT for business, 2011.

[10] E. Engström and P. Runeson, "Software product line testing – A systematic mapping study," Information and Software Technology, vol. 53, no. 1, pp. 2–13, 2011.

[11] "What is Risk Based Testing in Software testing?," Software Testing Class, 2014.

[12] "Risk Based Testing: Approach, Matrix, Process & Examples," Meet Guru99 - Free Training Tutorials & Video for IT Courses, 2019. [Online]. Available: https://www.guru99.com/risk-based-testing.html.

[13] G. Subrahmanyam, "The Ultimate Guide to Risk Based Testing: Risk Management in Software Testing", Softwaretestinghelp.com, 2018. [Online]. Available: https://www.softwaretestinghelp.com/risk-management-during-test-planning-risk-based-testing/. [Accessed: 22- Feb- 2019].

[14] Foidl, H. and Felderer, M. (2018). Integrating software quality models into risk-based testing. Software Quality Journal, 26(2), pp.809-847

[15] A. Hosseingholizadeh, "A source-based risk analysis approach for software test optimization". 2010 2nd International Conference on Computer Engineering and Technology, 2010

[16] S. P. Karmore and A. R. Mahajan, "New Approach for Testing and providing Security Mechanism for Embedded Systems," Procedia Computer Science, vol. 78, pp. 851–858, 2016.

[17] Wilson, F. (2018). How to Use the Risk Assessment Matrix in Project Management - DZone Agile. Retrieved from https://dzone.com/articles/how-to-use-the-risk-assessment-matrix-in-project-m-