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 Driven Approach to Test Device Software

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 Driven Approach to Test Device Software" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.87-92, **https://doi.org/10.31142/ijtsrd25230**

IJTSRD25230

ABSTRACT

Software testing is one of the most crucial testings in the software development process. Software testing should be scheduled and managed very effectively. The risk is the situation that has not occurred yet and may not occur in the future as well. After looking at this definition, risks can refer to the probability of the failure for a particular project. Risk-based testing is the type of testing that is based on the priority and importance of the software that has to be tested. In this research work, the new technique to test the device software has been proposed using the JAVA language. The new system is able to test the software based on various risks and provide alternatives based on that the risk can be reduced in the future. It also calculates the updated cost and duration required to complete the software when a risk has occurred. The proposed application is able to provide efficient and accurate results in terms of entered risks on the device software. In the future, the software can be used to test the device software for more number of risks to make it more suitable as per the user's requirements.

KEYWORDS: *Risk-based testing, Software Engineering, risk-based approaches, model-based testing*

I. INTRODUCTION

In the software product development process, it is very much necessary to reduce the cost and time that is required for the development of the software product. To achieve this purpose, the testing of the product at different stages of the development is very much required.

In the testing phase, the risk management and risk-driven approach play an important role in the software development so that testing of the product line can be efficient and effective. This approach is required to be applied to shorten the time and increase the productivity of the products [3].

Risk-Based Testing

A risk is referred to as a situation or problem which has not been happened yet and never happen in the future. It is defined as a possible problem. Risk-based testing is a software testing whose functions and features need to be tested based on importance, priority and potential failures. This type of testing is a best practice used for test analysis, estimation, planning, execution, design and result reporting. It is known as long-proven testing best practice. It is used for great agility. Its main purpose is to decrease the problems by using risk factor identification associated with software requirements. After the identification of the risks, the requirements are prioritized using risk analysis and strategies are designed for treatment of identified risks [5].

Risk-based testing involves the process of the testing of the software product on the priority basis by knowing the importance of the potential failures. The risk-based testing is required so that all the constraints that can affect the project like time, resources, quality can be taken care of in time [11]. For new projects, the risk factor is very much, therefore, it becomes the need to implement the risk-based testing in those projects.

The methods that can be used for the risk-based testing can include the clear understanding of the requirements of the product, documents related to design etc. and knowing about the views of the stakeholders related to the project. After knowing the requirements, it is important to prioritize the requirements, so that the planning and execution of the test case can be done in a better way [11]. Though the methods are known for this, still there is a need to formulate the proper framework so that the approach to the risk-based testing can be made easy.

Process of Risk-Based Testing

There are mainly five processes are involved in the complete process of the risk-based testing. These processes are risk identification, risk analysis, risk response, test scoping and test process definition [12].

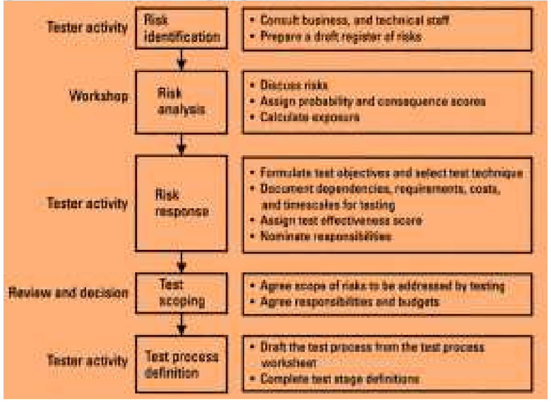

Tester activity
Risk identification
• Consult business, and technical staff
• Prepare a draft register of risks
Workshop
Risk analysis
• Discuss risks
• Assign probability and consequence scores
• Calculate exposure
Tester activity
Risk response
• Formulate test objectives and select test technique
• Document dependencies, requirements, costs, and timescales for testing
• Assign test effectiveness score
• Nominate responsibilities
Review and decision
Test scoping
• Agree scope of risks to be addressed by testing
• Agree responsibilities and budgets
Tester activity
Test process definition
• Draft the test process from the test process worksheet
• Complete test stage definitions

Figure 1: Risk-Based Testing [12]

Risk Identification: The risk is identified and then categorized, prepared a draft register of risks and sorting of risk is done to determine the significant risks. Its main goal is to determine the technical risks associated with requirements or software functionalities. It also includes a review of risk sources and to adopt the Taxonomy Based Questionnaire (TBQ).

Risk Analysis: In this process, all the identified risk is analyzed and determine the solution of those particular risks. It also prioritized software functionalities by using heuristic risk analysis. It informs values to metrics such as complexities.

Risk Response: This process involves the formation of the test objectives from the risks and selecting appropriate techniques in order to demonstrate test techniques or test activity in order to meet the test objectives. The test effectiveness score is also calculated by considering the document dependencies, cost, requirements and time required for software testing.

Test Scoping: It is a review activity which requires the participation of technical staff and stakeholders. It is important to adhere to the scope of risks. These risks are required to address by testing. All the members agree with the responsibilities assigned to them and the allocation of budget for these activities [12].

Test Process: Assumptions, test objectives and dependencies for each of the test stages have complied in standard format after finalized the scope of testing.

Advantages/Disadvantages

Some advantages of risk-based testing are that it helps in improving the quality. All the critical functions of the applications are tested. It gives more focus on the risks of the business project rather than the functionality of the information system. It also provides a negotiable instrument to test manager and client. In the testing process, test reporting takes places in a language which is understood by all the stakeholders. The testing process mainly concentrates on the important matters with optimaltest delivery such as qualifies resources, money and limited time. It also helps in improving the customer satisfaction, good reporting and progress tracking [9].

It helps in the prioritization of the tests so that deadlines for the particular task can be achieved. It leads to the better management of the project and resulted in the best usage of the resources that are related to the development of the software product. It provides the proper and efficient test coverage for the risk that can be associated with the product in the future. It will help in the reduction in the time and cost, by avoiding the risks in time, which will not impact the project in the future. A better and focused risk analysis can be achieved with the help of the risk-driven approach. There will be continuously monitoring of the project which will help in knowing the status of the project and quality will be maintained for the project [9].

Along with the advantages, there are disadvantages as well associated with it. There may be a case, where the risk that has been assessed is of very low intensity, and not need to be prioritized. This thing can lead to the wastage of the resources and the time. Though that risk needs to be assessed because it can cause a problem in the future, but it doesn't need to be prioritized and it can be taken care of later. This approach puts the focus on the risk identification, rather than having more focus on the development of the project. As the risk assessment will be on the subjective basis, therefore there can be other reality which sometimes, makes it unreliable. The identification of the stakeholders is important in this approach, but it is difficult to identify the right stakeholders for the project [9].

There is the need that the risk identification must be done in a proper way, so that the advantages part can be brought to use rather than, putting more stress on the disadvantages part. They need to real so that it can be assessed properly, rather than to be abstract which needs the more time for the assessment [9].

Role of Risk-based approach in testing Device software

It plays an important role in the risk management of the project. This approach is the most efficient way for the guidance of the project so that risks can be avoided in the future phase of the software development. This approach helps in the effective organization of the risks and then prioritize them according to their significance to the project. If there is the module, in which there are more than one risk, that module will be placed at the highest risk and resolving the issues of the module will be the priority. As the test cases are designed on the basis of the priority of the risk, therefore, there is a proper record for the risk identification and resolution of the risk, which enhances the quality of the project. It helps in reducingthe impact of the risk associated with the project in the future development [12].

The risk-driven approach is in its initial stage and can be improved further with the implementation of the available technology. The artificial intelligence is the buzzing word and technology these days, hence AI can be incorporated into the approach so that identification of risks can be done easily, and further time can be reduced, and the quality of the development can be increased. For the risk-based testing we first look out for the risks that may be associated with a project, then we analyse the risks that could pose harm to the potential costs of the projects.

Risks management help the team to find the components that are risky and amend them before the finalization of the project. There are various approaches for reducing the risk in software testing. Risk-based testing also allocates the various resources such as budget, time and persons for overcoming the various risks associated with the testing procedures that help in identification, analysis, and evaluation of the product risks.

II. RELATED WORK

According to **Hosseingholizadeh (2010),** software testing is one of the most important tests that are involved in the testing process for software. These are very expensive as well as they take up more than half of the total costs that are involved in a project. Risk management is one of the most effective techniques that can help in the management of the risks. They also help us in quality assurance for the product that is being developed. These risks management techniques also help to identify the components that may pose risks to the overall product. There are different techniques that are used for the risk management purpose, one of which is the determination of risks based upon the architecture of the components. These techniques use the software architecture that is applied before the implementation of the components. They can be used to identify the risky components and the same can be informed to the development team for the changes and alternations in the final product. In this paper, a source-based analysis is done. Source-based analysis helps to analyze the implemented software and find out the risky components in the final process. There are various advantages and disadvantages of this system. The advantage is that it determines the component even from the complex source code and the disadvantage is that they only identify the risky element without recording the efforts done for the same. Therefore, in this paper, the risk-based approach was presented with a hope to extend the research for identification of the risky patterns as well [15].

S. Karmore and A. Mahajan (2016) described new approaches for testing as well as also provide some security mechanism for embedded systems. The quality ofthe embedded system is one of the biggest challenges for the software testers all around the globe. Testing of one embedded system is different from another system. The approach which is used by the authors in this paper is basically used for testing the safety of the critical features of the embedded system. By using ANN, inputs and outputs are tested and validated. The security is also provided through critical classes and skipping invalid classes and by embedding secret key in RAM of the embedded device. The result indicated that the sensor activity is responsible for the success or failure of the embedded system. The new approaches used in this paper uses a classification method of artificial intelligence to generate three classes of conditions. Risk-based testing is performed to provide security to the embedded system by determining the risk regions. The proposed approach is effective in providing the security to the embedded system through the efficient testing mechanism. Also, the classification concept helps in the testing of a nonlinear embedded system [16].

According to **Foidl & Felderer (2018),** the testing approach which identifies the threats and vulnerabilities to software is said to be risk-based testing. Risk-based testing is the vital activity to assess the risks and this also supports the decision methods that are taken in the testing process. These models are an approved method for the quality assessment. In this paper, QuaMoCo quality model was used to check the risk and vulnerabilities to a software. The method is used in integration with a toolchain for assessing the risks and do the operations that are used in risk-based quality testing. In this paper, two approaches are used to show the quality assessment. First is the assessment of quality models in the risk-based models and second is the way those quality models will provide concrete integration to model QuaMoCo [14].

Trapani et al. (2015) proposed Prognosis and Health Management (PHM) approach for the risk assessment and for various risk methodologies as well. For the approach, they considered two case studies, that were based on this approach. Those companies used this approach for the manufacturing project of the same domains and ICT constraints. This research says that production and maintenance process are the main features in the life cycle of a project**.** The technical experts support the knowledge of Prognosis and Health Management (PHM) for the building of a project phase. The first company i.e. Company Alfa considered methodologies like HAZOP, FMECA, FTA for the risk assessment and these methodologies helped in manufacturing industries but they needed the planning phase as well as, they were not that mature in the business state. The second company i.e. Beta company skipped the planning phase as it is mature enough in the business state. The methodology is not upto the mark for the risk assessment and it is needed to be further efficiency in the approach [1].

Kolb (2003) deliberated the testing techniques and challenges in the software testing. For the more efficient and effective testing techniques, they proposed a new approach so that the product line can be tested in an effective and efficient manner. The aim of the approach was to reduce the costs, efforts, and time that is required for the testing of a product line. The focus of this paper is to plan the activities and design for testing in an efficient manner and to prioritize the resources and the quality measure for the risk and vulnerabilities testing. They considered the case studies to validate the approaches and experiments. The PULSE was the approach used to integrate the product line engineering i.e. completely customized. [3].

Fairbanks (2010), proposed a risk-driven approach that will work during the architectural design and will test the risks and examine its reduction to check that whether it is fit for the all the engineering disciplines or not. Their focus was to derive the risk management during the adoption of the model. They focused on the approaches that can help in lowering the risk and manage the risks efficiently. This approach favors in the decision making and overcomes the risk failures in the supportive manner [4].

This is found that from the literature reviews that most of the approaches are theoretical base approaches and they are not implemented yet, but they are providing the right direction for the testing techniques to deal with the risks and vulnerabilities in the product. So, that the plan should follow the software development life cycle to figure out the framework and it should be implemented at a large scale. There was a number of methodologies noticed in the previous studies and the further research will be done based on those studies so that the issues presented in that can be tackled easily.

III. PROBLEM FORMULATION

Software testing is a process in which we ensure that developed software is error free and performing as expected. In this process, we create test cases which have some predefined results. We use these test cases to test software by comparing predefined results with the results getting by running these test cases. Software testing is a very important phase of the software development life cycle. More than 70% time of the software development life cycle took this phase. The software cannot hundred percent bug-free but performing software test can reduce error form the software. Risk-based testing is a type of testing in which functions of the software are tested based on the priority, importance, and probability of the occurrence of a particular risk. In this approach, to test a software list of risk is prepared along with the various risk parameters (the priority, importance, and probability of the occurrence). Based on these risk parameters risk level or risk factor is calculated which is used in sort listing the test cases.

IV. IMPLEMENTATION

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.

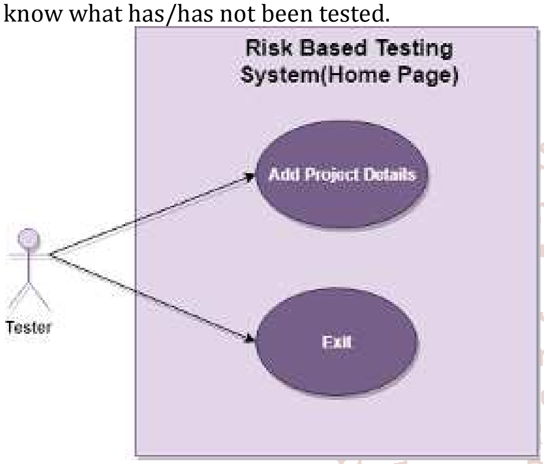


Figure 2: Use case of Home page for developed application

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.

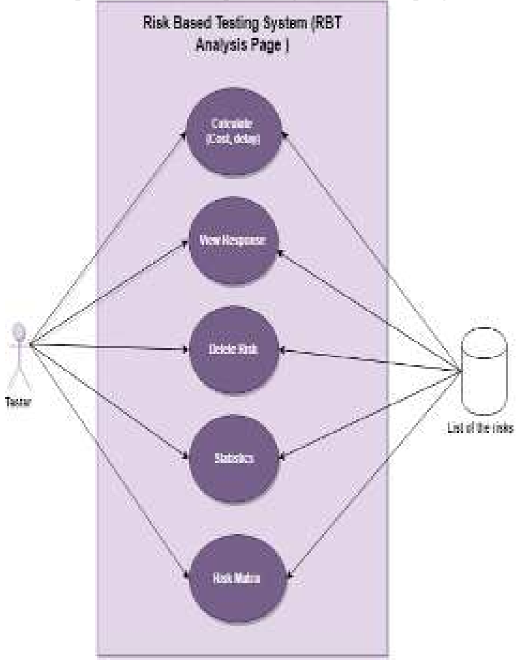


Figure 3: Use case of Risk-Based Testing application

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.'

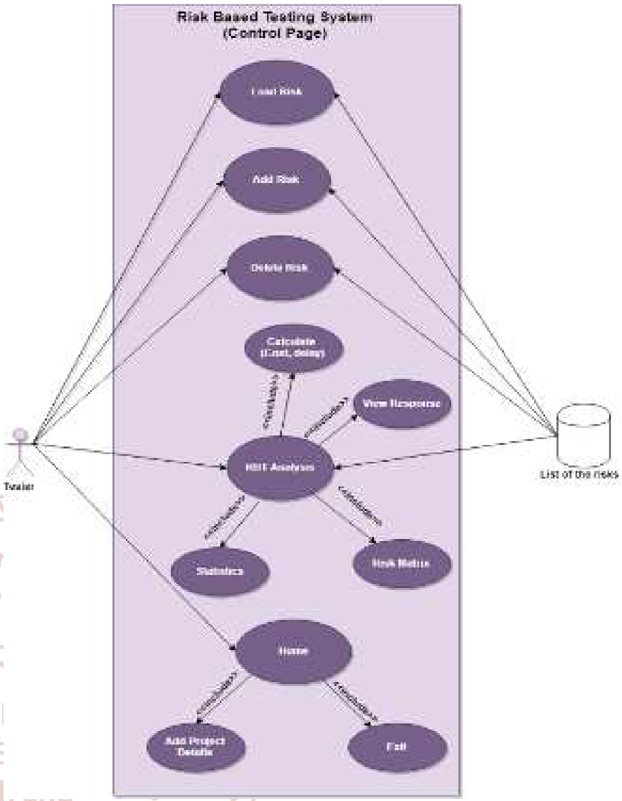


Figure 4: Use case for the control page of the application

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.

To categorize the risk based on their probability of occurrence and impact we have used the following conditions [17].

The probability of occurrence:

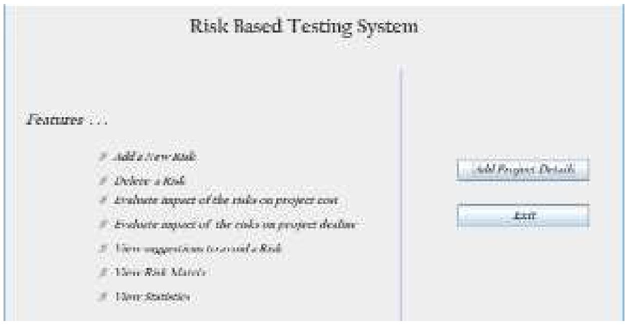
- 0 – 25: highly unlikely.
- 26 – 50: better than even (50/50).
- 51-75: probably and very likely.
- >76: almost certainly, highly likely.

To decide the impact:

- If the risk increases project cost up to 5% than the risk will be considered as a MINOR risk.
- If the risk increases project cost up to 5 - 10% than the risk will be considered as a MODERATE risk.
- If the risk increases project cost up to 10 - 20% than the risk will be considered as a MAJOR risk.
- If the risk increases project cost greater than 20% than the risk will be considered as a CRITICAL risk.

V. FINAL RESULTS

The results of the developed application are shown below. It includes the various options that are provided for testing the device software and gathering the required results so that accurate decision can be made in time without compromising the quality and time of the project.

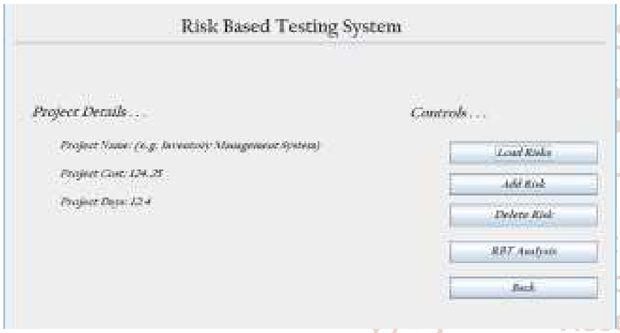

Risk Based Testing System

Features . . .

- Add a New Risk
- Delete a Risk
- Evaluate impact of the risks on project cost
- Evaluate impact of the risks on project deadline
- View suggestions to avoid a Risk
- View Risk Matrix
- View Statistics

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.


Risk Based Testing System

Project Details . . .

Project Name: (e.g. Inventory Management System)

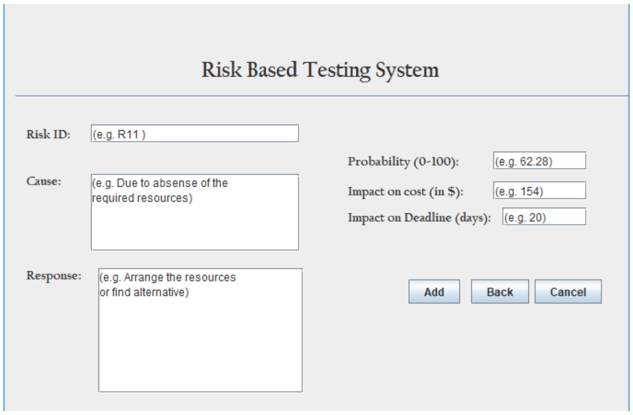
Project Cost: 124.25

Project Days: 12.4

Controls . . .

Figure 6: Control Page of Application

In the next screen as shown in Figure 6, the details of the project like project name, cost and time will be displayed. There are controls which will help in the analysis of the risks, as shown in the Figure like Load risks, Add Risk, Delete Risk and the RBT Analysis. As the tester will click on the Load Risks, the page will appear in which the details for the risk will be added as shown in the next Figure 7.


Risk Based Testing System

Risk ID: (e.g. R11)

Cause: (e.g. Due to absense of the required resources)

Response: (e.g. Arrange the resources or find alternative)

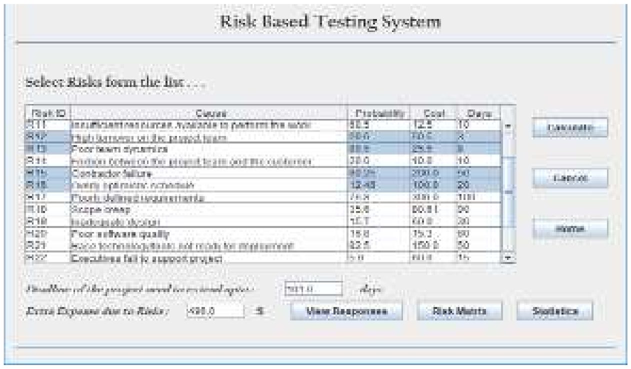
Probability (0-100): (e.g. 62.28)

Impact on cost (in $): (e.g. 154)

Impact on Deadline (days): (e.g. 20)

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, 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 Based Testing System

Select Risks form the list . . .

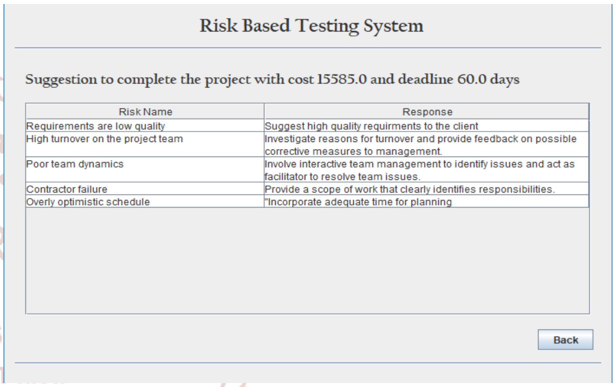
Risk ID	Cause	Probability	Cost	Days
R11	Insufficient resources available to perform the work	50.5	12.5	10
R12	High turnover on the project team	50.0	70.0	8
R13	Poor team dynamics	88.5	25.5	8
R14	Friction between the project team and the customer	30.0	40.0	10
R15	Contractor failure	60.25	2000.0	90
R16	Overly optimistic schedule	12.45	100.0	20
R17	Poorly defined requirements	74.8	300.0	100
R18	Scope creep	35.6	80.81	90
R19	Inadequate design	10.7	60.0	30
R20	Poor software quality	15.8	75.3	60
R21	Poor technology/tools not ready for deployment	82.5	150.0	50
R22	Executives fail to support project	5.0	[illegible]	15

Deadline of the project need to extend upto : 311.0 days

Extra Expense due to Risks : 498.0 $

Figure 8: RBT Analysis of risk

This Figure 8 shows the screen which will calculate the impact of the risk on the deadline and the cost of the project. On the selection of the risks from the table, the total cost and delay in the project will be calculated. On the click of the calculate button, the analysis of the risk will be done, and the measures will be provided for the risk avoidance.


Risk Based Testing System

Suggestion to complete the project with cost 15585.0 and deadline 60.0 days

Risk Name	Response
Requirements are low quality	Suggest high quality requirments to the client
High turnover on the project team	Investigate reasons for turnover and provide feedback on possible corrective measures to management.
Poor team dynamics	Involve interactive team management to identify issues and act as facilitator to resolve team issues.
Contractor failure	Provide a scope of work that clearly identifies responsibilities.
Overly optimistic schedule	"Incorporate adequate time for planning

Figure 9: Suggestions after the Risk Analysis

In Figure 9, the suggestion will be provided for the selected risks so that the risk could be avoided and there won't be any effect on the cost and deadline of the project. This application used to play an important role in the risk management of the project.

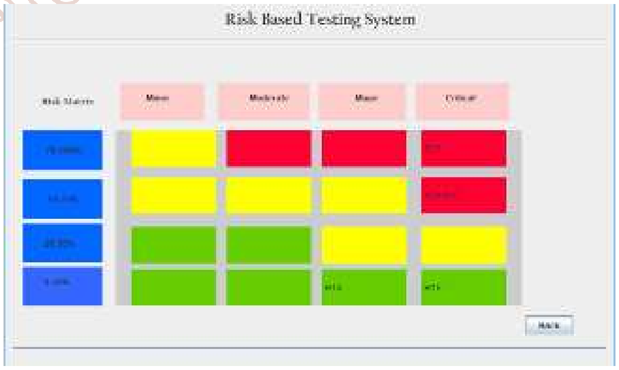

Risk Based Testing System

Figure 10: Risk Matrix

When user select risks from the risk table and want to see in which category a particular risk lies than user need to click the "Risk matrix" button. Selected risk-filled into different cells of the matrix based on the above conditions. More than one risks can be filled into the same cell.

This Figure 10 shows the Risk Matrix for the defined risks. On the basis of the analysis, the risk will be categorized as Minor, Moderate, Major, and Critical. This matrix also shows the percentage of the cost which can be impacted by the risk.

When user select risks from the table and click “Statistics” button it generates a graph which shows the graphical representation of the probability of occurrence, cost and impact on the deadline of the risk for each selected risk.

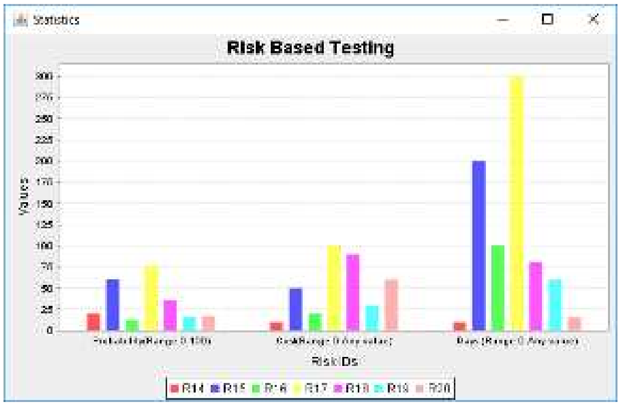


Figure 11: Graph for obtained results in risk analysis

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.

VI. CONCLUSION

The risk-driven approaches play a major role in the testing of device software. There is a need to be much conscious about the risk that is associated with the project so that in a later cycle of the software development, it should not affect the cost of the project very much. With this objective, the analyzation of the risk has become important in the software life cycle. This paper presents a new approach to test the device software using the risk-based testing. The proposed software in JAVA language is able to find the effect of the risk on the device software and also suggests the possible alternatives that can be taken to avoid or reduce that risk. The approach will help the testers to test the software based on various pre-defined risks and the user can also enter new risks in the system as well. In the future, the software application can be extended further to provide more options to test the device software like to provide options to manage software in parallel and to provide more options to manage and test the risks.

VII. REFERENCES

[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*, 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-1