Risk Driven Testing, Managing Risks and Quality Assurance: An Introspection and Benchmarking Implementation

Vinita Malik, Sukhdip Singh

Abstract*: The paper aims to critically analyze perspectives on Project Risk Management, Quality Assurance, Risk driven Testing and various computing environment possibilities in the field of risk driven testing. The research also makes a deep focus on methodologies, models and tools used in risk management, quality assurance and risk based testing. This paper utilizes an intelligent software vendor for establishing the correlation between risks management and quality assurance by benchmarking it against a particular technology. We have implemented the benchmarking of an Internet of Things application via smart software vendor for quantitative performance analysis of the competing products technology wise. All metrices of application get collected by application code scan and these are further donated to other organizations anonymously. These organizations can benchmark their own application as per technology, business drivers and application properties which in really helps in improving quantitative performance of the product. This research puts up a strong foundation for the advanced concepts related to risk driven testing, risk management, quality assurance and benchmarking of a software product.

***Index Terms*: Software Risk Management, Risk driven Testing, Quality Assurance, Benchmarking, Internet of Things**

I. INTRODUCTION

Statistics indicates that more than 53 % of software development projects have been behind budget and schedule and are unable to deliver features originally specified; 31 % of development projects are ended in premature cancelling or termination and around 61 % of them are able to satisfy original specified requirement features [1]. Project management associates itself with plan and management of resources or any type of changes in an organization in any dimension. The dimensions or area are staffing, products, people and services, production and distribution, budgeting, financing, purchasing, marketing, selling, human resources training and development or anything else which requires planning as well as management [2].The risk inherent in projects is required to be managed early in the SDLC model and testing of software is done based on risk identified and analyzed. The Software testing also needs selective and careful planning as software Testing is not performed exhaustively. The risky scenarios need proper identification of risky items because of quite restricted resources and shortage of time. Software quality is improved if early predictions about risk in software testing are done on time. Risk Management and software testing is joint together to solve two type of issues. Testing is mainly required to support the software risk management process and risk management to support software Testing [3].

The research has been organized to find out answers to following research questions:

RQ1: How we define Risk, Types of Risks, and Drivers for risks, Risk factors and Risk role in software testing process?

RQ2: Describe Project Risk Management, stages of Risk Management, approaches and models for Risk Management?

RQ3: Define Software Testing, Software Testing Life Cycle?

RQ4: Describe Risk driven/based Testing, various Approaches for Risk driven testing?

RQ5: Describe Computing Environment possibilities for Risk driven testing in the past?

RQ6: Describe Quality Assurance, Models, Methods and tools used in Risk driven Quality Assurance?

RQ7: Describe Benchmarking? How it is done by an intelligent software vendor?

We have explained all analytical perspectives in Section 2 related to Risk and Project Risk management. In Section 3, we have discussed about Software Testing, Risk driven Testing. Next section deals with the Computing Environments in Risk driven Testing. Section 5 talks about the quality assurance models and tools. In section 6, we have implemented the benchmarking of an Internet of Things software by an intelligent vendor for quantitative performance improvement of the IoT software. In next section, we will discuss and conclude about the research carried out about all analytical perspectives related to risk, risk management, risk based testing, quality assurance and benchmarking. In the last section, we discuss about future perspectives.

Search process has been executed manually by

Revised Manuscript Received on May 22, 2019.

Vinita Malik, Information Scientist, Central Library ,Central University of Haryana, Mahendergarh, India

Sukhdip Singh, Department of Computer Science and Engineering, D.C.R.U.S.T, Murthal, India

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

using the following search engines: Science Direct, Google Scholar, Springer Link, IEEE Explore, ACM digital library, IEEE Computer Society.

Extensive literature has been surveyed related to risk, risk management, risk oriented testing, software quality assurance, computing environment for risk oriented testing. Various kind of studies i.e. Modeling, Theoretical, review, simulation, case study, action research, experimental, empirical and exemplary have been explored to find out the answers for the research questions. The research is qualitative and descriptive in nature.

II. RISK MANAGEMENT

In this section, we have analyzed perspectives on project risks and risk management.

A. Risk

Risk characterization is done by two main factors i.e. Probability of occurrence and Severity of potential failure [4]. Probability of Occurrence is output of backward risks and consequence is output of forward risks [5].Risk is a futuristic harm that can arise from present actions taken. It comprises of uncertainty or lack of information [6] that can probably lead to loss [7]. Modifiers associated with risk are Risk proxy, Risk profile and Risk exposure [8]. Risk is one amongst the critical aspects in project management [9] and technological projects [10].Increase in the intensity of risk management strategies in the past years is because of socio economic pressures imposed by economics, politics and public awareness as software engineering without risk planning is riskier than ever [11].

B. Risk Types

Software risks are broadly categorized into following categories: Project life cycle risks, International, Supply Chain, Decision Making, Business, Structural, Reliability, Offshore, Outsourcing, Knowledge Transfer, Reverse Logistics, Safety, Economic, Quality, and Relational.

C. Risk Drivers

Following are the required drivers for risk in a software project [12]:
Software Availability: ability to deliver services whenever required.
Security: Accidental attacks are prevented.
Reliability: As per specification delivering the software services.
Safety: Operating software by protecting it from dangerous states.
Resilience: Recovering from unexpected faults timely.

D. Risk Factors

Risks factors are used for quantifying the Risk [12]. The risk factors may include Risk Exposure, Risk Proxy or risk profile.
Where Risk Exposure: =Risk Impact* Risk Likelihood

E. Risk in Software Testing

If risks are identified and assessed in the software testing life cycle phases i.e. planning of test cases, test selection, test prioritization, test designing, test coding and execution then software quality gets enhanced to a higher degree. Risks assessment is required to be done with effective planning so that project losses are minimized to a greater extent. For balancing the effort employed in the system and for avoidance of disasters risk probability and impact are predicted as accurate as possible.

F. Project Risk Management

Risk management is conglomeration of following four stages:
Identification of all risk factors, Analysis and assessment of various risk factors, To draw out plan for managing risks, Monitoring the risk plan and assessment strategies for continuous quality improvement.

G. Stages of Risk Management

Following are various stages of risk management:
Risk Identification: This is approached by collecting all information about resources i.e. project plan and risk plan once roles and responsibilities are assigned completely.
Approaches for Risk Identification: Following are the approaches by which risks can be identified in a software project:
Brainstorming, Checklist analysis, Cross functional teams, Root cause identification, Documentation reviews, Diagramming techniques, Delphi technique, Interviewing, Surveys, Force field analysis.
Risk Assessment: Risk assessment means how we are going to estimate impact and consequence of risk on project life cycle and taking care of overall accuracy in perceiving the likelihood of risk.
Approaches for Risk Assessment: Probabilistic approach, Decision tree and decision focus, Questionnaires, Risk metrics, Vulnerability metrics, Likelihood and loss expectancy, Impact and consequence, Risk exposure.
Risk Planning: Risk is planned either by avoiding or by making a contingency plan for minimizing the risk. Approaches for Risk Planning: Conducting market research, Risk pre-warning indexing, Changing technology, detailed study, Profit rate and Volatility rate, using consultants.
Risk Monitoring: Based on resource requirements risks such strategies aim to either reduce or transfer the risk exposure effect on software project. This type of strategies focus on how project risks are avoided or mitigated and further helps in project continuous quality improvement.
Approaches for Risk Monitoring: Industrial survey, Rough Set theory.

H. Risk Management Approaches

Following general approaches have been discussed for the risk management in the past [13]:
Check listing: Making a list of top risk factors is amongst the fast and low cost solution for risky items identification and assessment.
Analytical frameworks and Regions: Several frames target areas are defined in which risk management

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

methodologies are applied for reduction of risks. Risk Dimensions are used to frame risk regions.
Process Modeling: Under this strategy various models are used for proper sequencing of risks.

III. RISK DRIVEN TESTING

In this section, we study about software testing, risk based testing and quality assurance techniques.

A. Software Testing

The execution of an application with an intent of finding errors or bugs in the program is called as Software Testing [14].It is a process of verification and validation so that an application meets the all the expected business requirements. Testing is a complete process not a single activity. Software Testing has been found as second of three processes i.e. Production, testing, system use [13] (as seen in the Fig. 1).

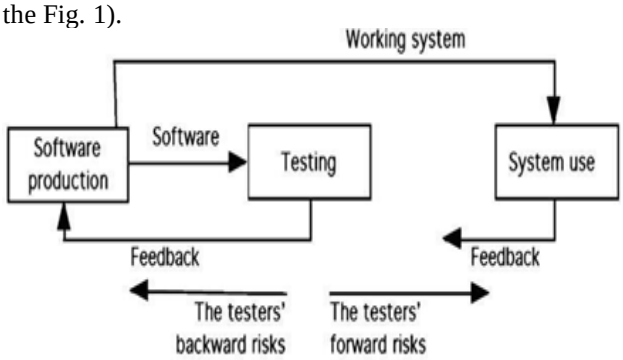

Fig. 1: Software testing as second of three processes

B. Testing Life Cycle

Standard test process of ISTQB has the following stages [15]: Planning of test cases, Analyze all test cases and designing them, Execution of designed test cases, Evaluation of Tests executed, closing all test case activities.

C. Risk Driven Testing

Risk driven testing takes risk as an inherent factor and works on its minimization. The project context must meet the given criteria before starting such kind of testing [16], [17]:
How the non-homogenous fault distribution be tackled according to severity levels, how risk gets coupled with different functions having different values, how component level dependency will be taken care of and what kind of technology will be used and how quality assurance will come into picture
The main objectives of risk driven testing are as follows [18]: Identifying all consequences, Severity level checking, Finding Faults, Doing Corrections and Reduction of Failures.

D. Risk Driven Testing Methods

Various risk driven testing methodologies have been proposed in the past.
Rosenberg [4] has proposed object oriented scenario in which more risky classes gets identified first which have more chances to fail. Redmill[5] has integrated risk management and testing process to make testing more productive. He concentrated on analysis of single factor that may be Probability or impact and both in combination. Felderer and Ramler[16] has focused on integrated risk based testing . They talk about an empirical study for integrating risk imbibed testing into already existing test process. Yoon and Choi[19] has pro- posed technique for test cases prioritization for risk driven testing and calculated its effectiveness in a traffic model. Stallbaum[20] has proposed methods for risk oriented test case prioritization by the help of activity diagrams. He prepared a prototype for his approach and then applied them in a software product. Zech [21] proposed security modeling in the cloud computing. Misuse case approach has been used due to large number of test cases in cloud environment. Gebizli[22] has talked about a risk oriented testing by Markov chains modeling by which system states transitions gets assigned with probability values.
Amland [23] talks about a risk driven testing in which risk prioritization is done to resolve high risk areas. He has applied his approach on banking systems. He also focused on how flexible organization of team may help in finding bugs easily. In the Fig. 2 it has been shown clearly.

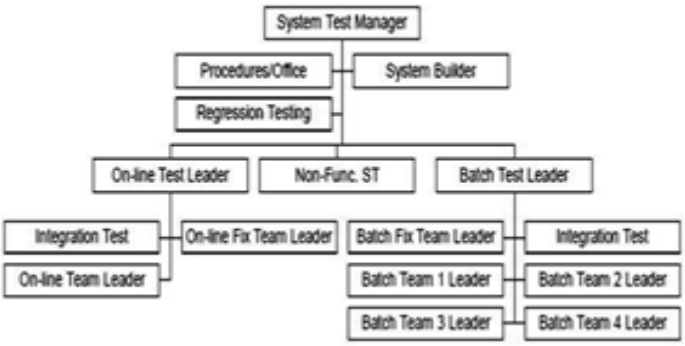

Fig. 2. Software Team arrangement as per risk assessment

Automated risk driven is also done by behavioral modeling to extract the safety information and then creating risk driven test models [24]. J.Bach [25] proposed heuristic risk driven testing with two approaches first was inside out and second one was outside in. In first one the testing team starts by the risk identification to find out vulnerabilities in the system. But in the second approach testing team only works with visible risk items at present. Chen [26] has talked about an approach for optimization of regression based test cases. He has found out a specification relied method for selecting test cases. P. Gerrard [27] has worked on risk driven testing by failure mode and effect on business risks. Jogensen [28] developed a tool for preparing test plan in risk driven environment. Veenendaal [29] used practical risk based testing approach (PRISMA) for differentiating between business and technical risk factors once overall risks of software were calculated. Raparla [30] has used a tool known as QUART-ER for project risk assessment. Zimmermann et. al. [31] has devised a method for safety systems formal modeling.
Kloos et.al. [32] Used failure tree analysis for risk assessment in

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

combination with Melay machine.

IV. COMPUTING ENVIRIONMENTS FOR RISK DRIVEN TESTING

Following are the various computing environments for risk driven testing:

A. Computing Environment: Cloud Computing

Various layers i.e. Service Layer, Infrastructure and Platform Layer in cloud computations need security. Cloud Consumers develop customized applications and providers offer cloud as a service. Negative requirements are derived by risk analysis under test [21].The UML is used for negative requirement modeling. As seen in the Fig. 3. It has been depicted how risk driven testing is done in cloud environment. Security risks in cloud computing has been handled by various algorithms like split algorithms in past [33].

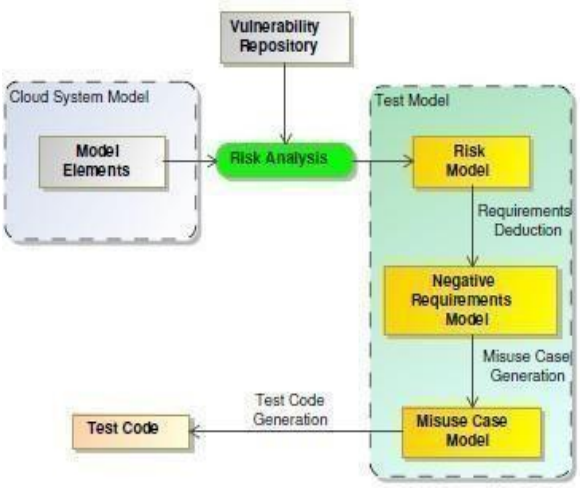

Fig 3. Cloud Computing Environment for risk driven test cases

B. Object Oriented Computing Environment

In this, metrics are used for the risky class identification. The metrics involved are #methods, method/class and response / class, tree depth and children count. The purpose is to identify high risky classes [4].

C. Service Oriented Computing Environment

Services to be tested included IT related or web services. For IT services we may perform Risk Profiling and for Web based services ontology based risk evaluation is done [34].For each ontology class we calculate its failure probability and then try to adjust its estimation. We try to analyze risks at three layers i.e. Data, Unit or Integration Layer. Control flow analysis is done for whole system. Services are tested at data, unit and integration layers [35]. Complexity constructs are drawn as seen in the Fig.4 shown below:

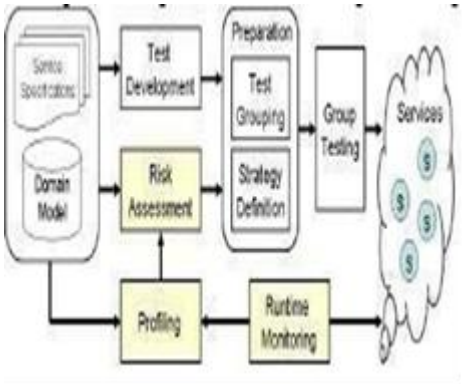

Fig. 4. Adaptive services Testing

D. Industrial Computing Environment

In the industrial risk based evaluation risk exposure is considered as main metric. Volatility of requirements and coding complexity are to be considered mainly for test case prioritization. Following metrics are considered mainly i.e. Prioritization value, Severity and weighted priority [36]. The abnormal behavior of the system is also captured by misuse cases [37].

E. Network Computing Environment

The networking artifacts are modelled by safety and availability parameters. Network is comprised of various components having common or different control [38]. With fault tree analysis test case generation is possible for a networked system as shown in the Fig. 5.

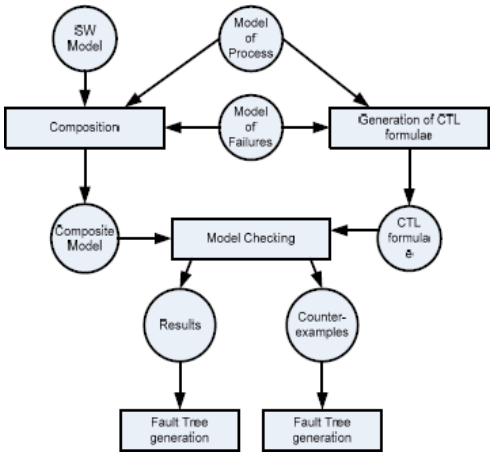

Fig. 5. Fault Tree Analysis in Networks

F. Smart Computing Environment

As the smart environment or ubiquitous environment is highly evolutionary in nature so such type of computing is really complex as user intent, energy requirements; system handling requires high proactivity [39],[40].This computing requires combination of both mobile as well as distributed computing for considering system risk levels. Fuzzy techniques has also been used in past for disease estimation smartly by taking health risks into account [41].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

V. QUALITY ASSURANCE

For assuring system quality risks need to be reduced by techniques like reviews, model checking and inspection as these consume most of the resources of a software project .We require to do right planning for right quality assurance. All the project panning should be effective to handle the risk. So expected number of faults is required to be found
out to plan QA activities [42].Quality assurance is one of the most essential aspect in software quality management where the quality management talks about all activities to meet the life cycle implementations[43] .Quality management of software includes that are quality planning , assurance of software quality and control on quality[44].

A. Quality assurance Methods

In the embedded systems Quality is done with the help of metrics. Integrated (Static + Dynamic) quality assurance is utilized to produce knowledge for improving software quality. Such techniques include defining context, calibration of quality assurance, Prioritization and evaluation of assumptions and packaging [45].In the web testing environment, we apply Cooperative agent approach for quality assurance [46]. One more method is QATAM means Quality Assurance Trade off Analysis Method for identifying and evaluating QA strategy [47]. Another method includes quality profile building which is a strategy matrix for telling quality characteristics. The communication with users is also on the topmost priority [48].

B. Quality assurance Models

The father of quality assurance model is considered as Mc Call who introduced the concept for US Army software development project. He considered Product revision, transition and operation as key values [49]. QuaMoco model was introduced to operationalize ISO/IEC 25010 model. Wager divided quality models into two types software quality composition factors and quality models characteristics. Product factors used to define the traceability matrix for quality characteristics. Overall quality metrics are formulated for building the taxonomy for assurance models [50]. Dromey proposed a QA model with quality components and components interrelationships which posed impact on software properties too [51].

C. Quality assurance tools

Many tools have been utilized in past for assuring the software quality. FOCUST was developed for failure pattern classification in which relation with the defect classes are defined. Scenarios are developed for the user as well as tester [52]. Risk based analysis is also done by a language known as TTCN-3 used in embedded systems mainly. The framework jABC is used for creating system automata for the service oriented architecture. All the reusable components are used in the form of a graph structure [53].The architecture conformance analysis uses Find Bugs 1.3.9 for finding bugs in java based system [54].We have j-Learn as an important tool for integrating testing and quality assurance [55]. Another framework is X Query based analysis framework which is quite flexible in nature and supports automated analysis of software artifact.

VI. IMPLEMENTATION RESULTS FOR BENCHMARKING OF AN IOT APPLICATION

Benchmarking does the quantitative analysis of the competitive products of the market [56].The good benchmark must imbibe some application dimensions and the specification should be easily understandable, with minimal hardware dependencies [57].The benchmarking is only employed after well convergent with knowledge of software product design features and h/w s/w environments [58].In this research, we have done the benchmarking of an IoT software i.e. NodeRed, where the software source code is downloaded by the Github repository [59]. The benchmarking has been done by the intelligent vendor [60] on basis of code technology and properties. The code is scanned by intelligent vendor taking all metrices which are further donated anonymously to various customers across the globe. Those industries or organizations use these particular metrices to benchmark their own application as per technology and properties. The application properties have type of application i.e. custom application/COTS, year of application implementation and application user name. The technology type indicates the technology which that particular application supports. Various business drivers and application properties are considered for benchmarking as given in the Fig. 6 below:

Fig. 6. Application Properties: Nodered application

The values considered for business drivers are taken as shown in the Fig. 7:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

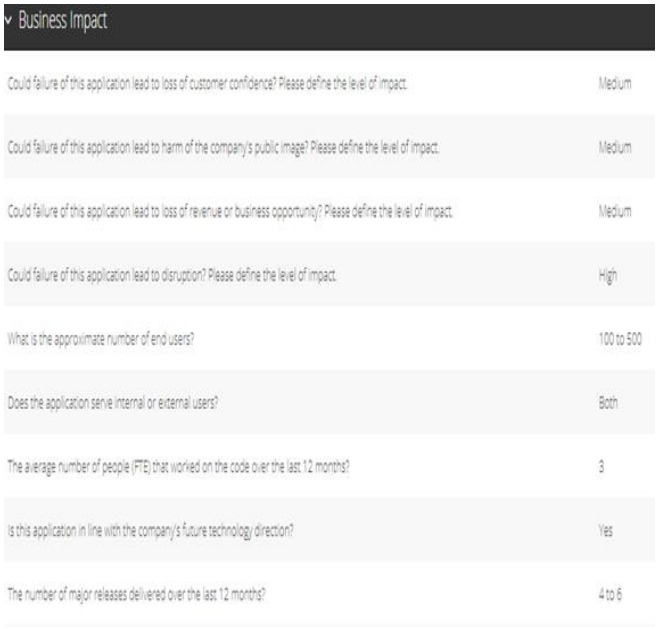

Fig. 7. Metrices for Business Driver 1: Nodered application

In the Fig. 8, the internet of things application supports python. So the benchmarking is done for that particular technology.

Fig. 8. Benchmarking against python technology

The black dot represents the IoT application benchmarked for python technology is in the second quadrant representing the application belongs to upper 50% of the applications if measured for software resiliency to make the bullet proof software[61]. The yellow dots indicates the inside organization applications and grey plus represents the outside applications. The benchmarking for Ksh technology is given as below in the Fig. 9.

Fig.9. Benchmarking for Ksh technology

In the Fig. 10, the benchmarking is done for JavaScript technology.

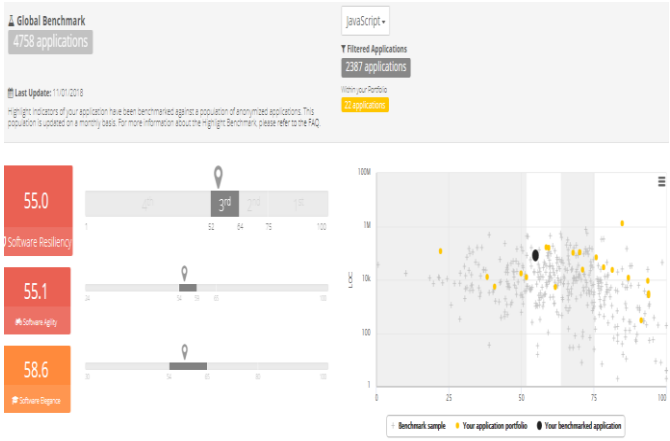

Fig. 10. Benchmarking for JavaScript

In the Fig. 11 and 12, the benchmarking is done for various business and technical drivers.

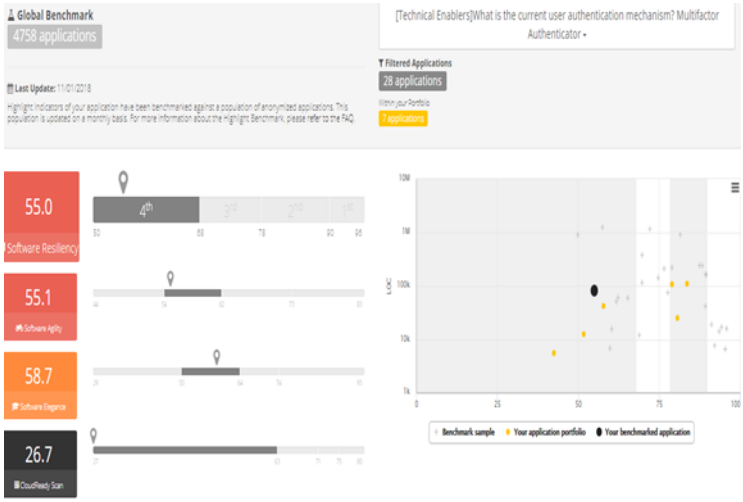

Fig. 11. Benchmarking for Technical Driver 1

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Again the black dot is in 2nd quadrant in the Fig. 11 and 12 indicates that the software resiliency of the application belongs to bottommost 25% applications as benchmarking is done against technical driver 2.

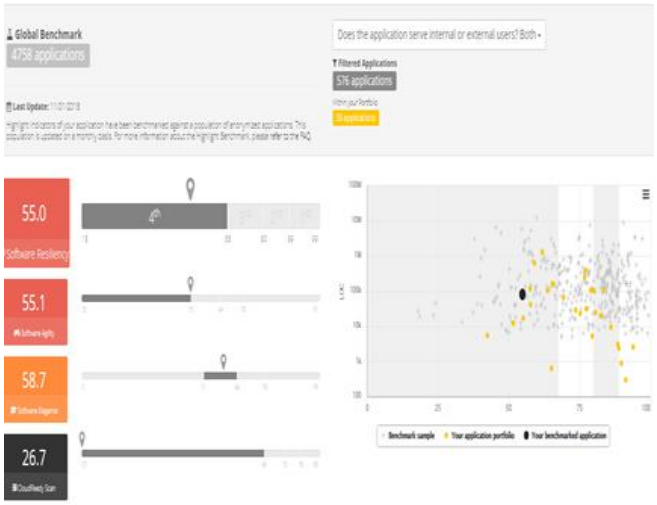

Fig. 12. Benchmarking for Technical Driver 2

VII. CONCLUSIONS

As we have described in detail about Risk, Risk Management, Risk driven testing and quality assurance and analyzed how much it is essential to consider risk in the initial stages of the software project development. The stakeholder's interaction is more and customer communications gets easier. In risk driven testing test prioritization helps in reduction of stabilization costs which helps in quality assurance. Identifying and Selecting right stakeholders are really a difficult task. Following are the conclusions for our research questions.

RQ1: How we define Risk, Types of Risks, and Drivers for risks, Risk factors and Risk role in software testing process?
A1: As described in Section 2.
RQ2: Describe Project Risk Management, stages of Risk Management, approaches and models for Risk Management?
A2: As described in Section 2.
RQ3: Define Software Testing, Software Testing Life Cycle?
A3: As described in Section 3.
RQ4: Describe Risk driven/based Testing, various Approaches for Risk driven testing?
A4: As described in Section 3.
RQ5: Describe Computing Environment possibilities for Risk driven testing in the past?
A5: As described in Section 4.
RQ6: Describe Quality Assurance, Models, Methods and tools used in Risk driven Quality Assurance?
A6: As described in Section 5.
RQ7: Describe Benchmarking? How it is implemented by an intelligent software Vendor?
A7: As described in Section 6.

VIII. FUTURE SCOPE

This research has focused on various tools, technologies, methodologies, models related to software project risks, Risk management and risk driven testing to explore the possible aspects for software quality assurance models, methods and tools. We have implemented the benchmarking of an IoT application by an intelligent vendor by taking software agility and resiliency into consideration on the basis of the supporting technology and application properties. The quantitative performance of the software is enhanced and we also get to know to by what extent we need to make our software bullet proof. In future, the risk based testing may be implemented in evolutionary environments for establishing correlation of risk based testing and quality assurance.

REFERENCES

1. Miguel Wanderley, Julio Menezes , Cristine Gusmao, Filipe Lima , "Proposal of Risk Management Metrics for Multiple Project Software Development", International Conference on Enterprise Information Systems, vol. 64, pp: 1001-1009, October 7-9, 2015
2. Tanzeem Bin Noor, Hadi Hemmati, "Test Case Analytics: Mining Test Case Traces to Improve Risk-Driven Testing", IEEE 1st International Workshop on Software Analytics (SWAN), pp: 13-16, 2015
3. Gencer Erdogan ,Yan Li, Rsgnhild Kobro Runde , "Approaches for the Combined use of Risk Analysis and Testing: A Systematic Literature Review", International Journal of Software Tools Technology Transfer, vol 16, pp 627-642, 2014
4. Linda H.Rosenberg ,Ruth Stapko , Albert Gallo, "Risk-based object oriented Testing, Proceeding of the 24th Annual Software Engineering Work- shop", . NASA, Software Engineering Laboratory, pp. 1–6,1999
5. Felix Redmill , "Exploring risk-based testing and its implications", Softw. Test. Verif. Reliab,vol.14,pp: 3–15, 2004
6. Gerald F. Smith ,P. George Benson, Shawn P. Curley, "Belief, Knowledge and uncertainty: A cognitive perspective on subjective Probability", published in Organizational Behavior and Human Decision Processes, vol 48, pp:291-321,1991
7. https://www.castsoftware.com/research-labs/code-quality-application-quality, downloaded on October 2018
8. https://mpra.ub.uni-muenchen.de/61329/, 2015
9. A. Sivathanu Pillai ,A. Joshi , K. Srinivasa Rao, "Performance measurement of R&D projects in a multi-project, concurrent engineering environment", International Journal of Project Management ,Issue 2,vol. 20, pp: 165-177, 2002
10. Henry Barki , Suzzane Rivard, Jean Tablot, "An integrative contingency model of software project risks management" ,Published in Journal of Management Information Systems ,vol. 17(4),pp: 37-69,2001
11. A.Fuller ,P. Croll ,O.Garcia, " Why software Engineering is riskier than ever", IEEE Comput. Soc Second Asia-Pacific Conference on Quality Software - Hong Kong, China, Proceedings Second Asia-Pacific Conference on Quality Software,pp:113-119, 2001
12. Michael Felderer, Ina Schieferdecker , "A taxonomy of risk-based testing", Int J Softw Tools Technol Transfer , vol.16, pp:559–568, 2014
13.Paul.L. Bannerman, " Risk and risk management in software projects: A reassessment", The Journal of Systems and Software, vol. 2, Issue:2, pp:2118-2133, 2008
14. http://istqbexamcertification.com/what-is-software-testing/
15. ISTQB. Standard glossary of terms used in software testing. Version 2.2. Brussels: International Software Testing Qualifications,2012
16.Michael Felderer ,Rudolf Ramler, "Integrating risk-based testing in industrial test processes", Springer Science+ Business Media, Software Qual J, vol. 22, pp:543–575, 2014
17. Rudolf Ramler , Stefan Larndorfer , "What software repositories should be mined for defect predictors",35th Euromicro conference on software engineering and advanced applications ,pp: 181-187, 2009
18. Felix Redmill, "Theory and practice of risk-based testing", Softw. Test. Verif. Reliab., vol. 15, pp:3–20 , 2005
19. HoijinYoon , Byoungju Choi , " A test case prioritization based on

degree of risk exposure and its empirical study", Int. J. Softw. Eng. Know. Eng., vol.21(02), pp:191–209, 2011

20. Heiko Stallbaum ,Andreas Metzger, Klaus Pohl, " An automated technique for risk-based test case generation and prioritization", proceedings of the 3rd international workshop on automation of software test. ,pp. 67–70, 2008

21. Philipp Zech , " Risk-based security testing in cloud computing environments , Proceeding of the Fourth International Conference on Software Testing", Verification and Validation (ICST‟11), pp.:411–414,2011

22. Cereri Sahin Gibizli, "Combining Model based and risk based testing for effective test case generation", IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW),2015

23. Stale Amland, "Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study", Syst. Softw.,vol. 53(3), pp:287-295,2000

24. Ramy Nazier, Thomas Bauer , "Automated risk-based testing by integrating safety analysis information into system behavior models", Proceedings of the 23rd International Symposium on Software Reliability Engineering Workshops (ISSREW‟12), pp :213–218. IEEE, New York, 2012

25. James Bach, "Heuristic Risk-Based Testing. Software Testing and Quality Engineering Magazine", pp: 96-98, 1999

26. Yanping Chen, Robert L. Probert, and D. Pual Sims, " Specification-based Regression Test Selection with Risk Analysis", Proceedings of the conference of the Centre for Advanced Studies on Collaborative research (CASCON 02), pp: 1.,2002

27. Paul Gerrard , "Risk-Based E-Business Testing", ISBN 1580533140, pp: 51 – 80, 2002

28. Lars Kristoffer Ulstein Jogensen , "A software tool for risk based testing", Graduation work, Norwegian University of Science and Technology Norway,2004

29. Erik Van Veenendaal, "Practical risk-based testing: the PRISMA approach", UTN, Cambridge, 2012

30. Hima B. Puppala Raparla R. , Linda B. Sherrell ,A Tool for Risk-based Testing , 2007

31. Fabian Zimmermann, Robert Eschbach ,Johannes Kloos ,Thomas Bauer , "Risk- based Statistical Testing: A Refinement-based Approach to the Reliability Analysis of Safety-Critical Systems", In: Proceedings of the 12th European Workshop on Dependable Computing (EWDC), France, 2009

32. Johannes Kloos, Tanvir Hussain, Robert Eschbach, "Risk-Based Testing of Safety-Critical Embedded Systems Driven by Fault Tree Analysis", Proceedings of the IEEE Fourth International Conference on Software Testing, Verification and Validation (ICST) IEEE Computer Society, Berlin, pp:26 33, 2011

33. Priyansha Garg. Moolchand Sharma, " Security on Cloud Computing Using Split Algorithm Along with Cryptography and Steganography", International Conference on Innovative Computing and Communications, Lecture Notes in Networks and Systems 55,pp: 71-79,2019

34. Jilene L. Sauve , "Risk based Service Testing, Several approaches to the application of risk to IT Service Management, IEEE Xplore , pp:106-109, 2008

35. Xiaoying Bai, Ron S. Kenett , Wei Yu, "Risk assessment and adaptive group testing of semantic web services", Int. J. Softw. Eng. Knowl.

Eng. vol. 22(5), pp.595–620, 2012

36. Hema Srikanth , "Requirements based test prioritization using risk factors: An industrial study, Information and Software Technology", vol.69, pp:71–83, 2016

37. K.Krishna.Murthy, Kalpesh R.Thakkar, Shirsh,Laxminarayan, "Leveraging risk based testing in enterprise systems security validation", Proceeding of the First International Conference on Emerging Network Intelligence (EMERGING‟09), IEEE, New York, pp:111–116, 2009

38.Tanvir Hussain, "Automated fault tree generation and risk based testing of networked automation systems", IEEE, 2010

39. M.Satyanarayanan , "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, 2001

40.Lathies Bhaskar , "Pervasive Computing issues, applications and challenges", International Journal of Engineering and computer science

,vol. 2,pp :3337-3339, 2011

41. P. Shrivastava P., N.Sharma, "Fuzzy Risk Assessment Information System for Coronary Heart Disease", International Conference on

Innovative Computing and Communications, Lecture Notes in Networks and Systems vol.56, pp :159-170, 2018

42. BedirTekindorgan, Ali Nour, Ivan Mistrik, "A chapter on quality concerns in large scale and complex software intensive systems", Elsevier,pp:1-17, 2016

43. Norbert Greif , " Software Testing and preventive quality assurance", Computer standards and interfaces, Elsevier, vol. 28, pp. : 286-296,2005

44. Michael Klas,Haruka Nakao, " Support planning and controlling of early quality assurance by combining expert judgment and defect data A case study",In journal of Empirical software Engineering ,vol. 15, issue 4 , pp:423-454,Springer verlag, 2010

45.Frank Elberzhager ,Alla Rosbach ,Thomas Bauer, "An Integrated Analysis and Testing Methodology to Support Model-Based Quality Assurance", SWQD ,Lecture Notes in Business Information Processing, vol 166. Springer, Cham, 2014

46. Hong Zhu, "Cooperative Agent approach to quality assurance and testing web software", Proceedings of the 28th Annual

International Computer Software and Applications Conference (COMPSAC‟04), vol. 2, pp:110-113, 2004

47. Dietmar Winkler, Frank Elberzhager, : "Software Process Improvement initiatives based on Quality Assurance Strategies: A QUATAM Pilot application", CCIS, 2010

48.http://www.ascd.org/publications/books/105006/chapters/ Understanding-the-Quality-Profile.aspx

49.Mario Gleirscher, " Introduction of static quality analysis in small and medium sized software enterprises: Experiences from Technology

Transfer", Springer International Journal of Science+ Business Media, 2013

50.Harald Foidal ,Michael Federer, " Integrating software quality models into risk based testing", Software quality journal, Springer

Verlag,vol 26, pp:809-847,2018

51. BedirTekinerdogan ,Ali Nour, John Grundy , "Quality concerns in large scale and complex software –intensive systems", Software quality assurance, Elsevier,pp:1-17, 2016

52. Konstantin Holl,Frank Elberzhanger, " Towards a perspective based usage of Mobile failure patterns to focus quality assurance", Springer International Publishing, 2015

53. Johannes Heinrich Neubauer, " Risk based testing via active continuous quality control", International Journal of software tools technology ,

2014 Transfer", Springer International Journal of Science+ Business Media, 2013

54 Alexander A.Hernandez , "JLearn: An Instructional Environment for Java Program Composition Integrating Test Driven Development and

Life cycle management for software Quality Assurance", International Conference on Networking and Information Technology,2010

55. Jens Nodler , Helmut Neukirchen, "A flexible framework for quality assurance of software artifacts with applications to java, UML, and TTCN- 3 Test Specifications , International Conference on Software Testing Verification and Validation, 2009

56. Thomas M. Conte, Wen-Mei W. Hwu, "Benchmark Characterization", Computer ,vol.24,pp:48-56,1991

57. Jim Gray,"The benchmark Handbook; For databases and transactions processing systems,"Morgon Kaufmann Publishers ,Inc. San Mateo, California ,pp:1-15,1991

58.S. Madan,S. (1997), "A benchmark for the artificial intelligence applications on parallel computers –BEAP", Conference on Communications, Power and computing, WESCANEX 97 Proceedings; Winnipeg ,pp: 82-87,1997

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

59. https://codeload.github.com/node-red/node-red/zip/master,October 2018
60. https://www.castsoftware.com/products/highlight
61. https://doc.casthighlight.com/software-resiliency/

AUTHORS PROFILE

First Author Ms. Vinita Malik received the Bachelor of Engineering degree from the Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Haryana, India in 2008. She got her Masters of Software systems degree in 2012 from B.I.T.S Pilani Goa Campus. She is currently posted as Information Scientist at Central University of Haryana, Mahendergarh. Her research interests include risks management in various computing environments in software engineering. She has published more than 15 papers in the Journals and conferences of repute.

Second Author Dr. Sukhdip Singh received the Bachelor of Technology degree from the Maharishi Dayanand University Rohtak , Haryana, India in 1999 and the Ph.D. degree from Maharishi Dayanand University Rohtak ,Haryana India .He worked as a lecturer in Technological Institute of Textile Sciences Bhiwani Haryana from 2000-2002. He is currently a Professor in the Department of Computer Science and Engineering at Deenbandhu Chhotu Ram University of Science and Technology Murthal, Sonipat Haryana India . His research interests are in Software Engineering, Green Computing and Cloud Computing. Dr. Singh has published more than 35 papers in the Journals and conferences of repute. He serves as advisory committee member in various conferences, He is a life time member of the ISTE .

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication