www.ioirp.com

International Journal of Innovative Research in Technology, Science & Engineering (IJIRTSE)
ISSN: 2395-5619, Volume – 1, Issue – 6. August 2015

Qualitative &Quantitative (Q&Q) Risk Resolving Technique based on Framework View - Risk's Finding and Solution

[1]J.DILLIBABU, [2]DR (MRS) K.NIRMALA
[1]ASSISTANT PROFESSOR, St. Thomas College of Arts & Science, Chennai, India
[2]SR.LECTURER, Quaid-E-Millath Govt College (w), Chennai, India
[1]jdbabumca@rediffmail.com, [2]nimimca@gmail.com

***Abstract*— Software is a combination of internal components binded with one or more function's to perform a task under defined, undefined conditions. It also a combination of Technical Functional Units, These functional units perform well and some times may not give the expected resultant values for the End user's. This is because of logical and Requirements diversions modified by too poor quality of requirements and unidentified behavioural values processed by those system, Normally Systems can built under the shadow of Umbrella activities it is said to be Framework Activities, some processes such as Waterfall Model, Spiral Model, Prototype model, etc., These model process based on standard requirements collected by the Analysis team followed by Design Quality also includes Implementation Standards existing in an Organization. Before stepping into these empirical activities we need a standardisation awareness, Software Development Phase (SDP) Risk Finding and Management approaches. Risks are two categories Qualitative and Quantitative (Measurable and Non-Measurable) in these two risks groups possibility of identifying Qualitative risk's is more Sensitive these risks will take more time complexity, Data Complexity, Functional Complexity as reflective behaviour. In this paper we found some Frame work tool in the way of Simulation tool (ST), Boundary covering (BC), etc. These frame work gives a standard solution for both types of Risk's also we explores technical conclusion for the risk categories.**

Keywords*— *Waterfall Model, Spiral Model, Prototype Models, ST (Simulation Tool), BC (Boundary Covering).

I. INTRODUCTION

Software project suffers from many problems like high computational cost, higher delay time in designing theProjectsthese problems are solved using the software risk management which helps the software developer to identify, analyse and it deal with software risks items. Software risk management is also an attempt to define and formulate the risk oriented connection of success into a definite set of methods and techniques. SoftwareIndustries concentrating more on project complexity with Different strategic management decisions will be implemented to create different sets of risks with different cost commitments. Each strategic decision requires a project management plan with a unique budget and schedule of software development.Computer-Aided Software Engineering (CASE) toolprovides programmed support for software development time.The aim of CASE tools is to reduce the delay time andcomputational cost of software development process toenhance the quality of the systems. CASE tool gives a flexibility to increasing productivity, improving quality of the product, improving the maintenance Project Planning and Control aims to

- Handling the risks related to software planning and control by the software engineers in order to develop risk-free software products.
- Software project complexity reduction process.
- To Generate Expected message Alerts also to improve the quality of software risk Controls.

This paper contains four segments I. Introduction give brief intro about Frame Work Tools. II. Simulated Framework and its operational Behaviour. III. Software Maintence Risk monitoring is technical approach help us to identify risk and its Severity levels. IV. Branch Coverage V. Automated Risk Identification approach view. VI. Statistical Comparative Data Representations.

II. SIMULATED FRAMEWORK TOOLS

The strategic management process is about designing and undertaking decisions of strategic importance in order to manage and control strategic project parameters. Strategic Management Process (SMP) [1], simulating the impact of planned types of decisions namely, cost, risk, budget requirement and scheduling of the project using simulation model .It also had the advantage of an integrated framework where several planned factors were integrated to identify the risk and cost earned during the design of project to provide a critical. Scope of strategic management focuses on the overall management of projects and defines a sets of directions to the project development activities, whereas multiple strategic options used for a specific software project development, Parameters quantifying using simulations. Strategic decisions highlights the implications of each strategic decision on the project management plan need's a methodologies for Simulation, this model provides a ways to examine the consequences of strategic decisions on strategic project parameters and project management plans to help software organisations needs to select the best strategic decision that supports their resources such as budget, risk tolerance and management style. Simulation model for the strategic management for software. Simulation model emphasises the effects of strategic decisions on strategic parameters of software projects and explains how these affect project management plans. Simulation model is based on System dynamic Technique (SDT) and Discrete Event Simulation (DeST) presents a model based on the analysis of risk and cost to support strategic decision making in software development projects with two approaches one Data could be importing in earlier project, Two Decision making also done on the cost-effectiveness different risk mitigation alternatives. Simulation models majorly deals with planning, control, improvement and training of software development processes. SMP Risk management of software development projects involves set of activities like Identification, analysing and impact of risk management events.

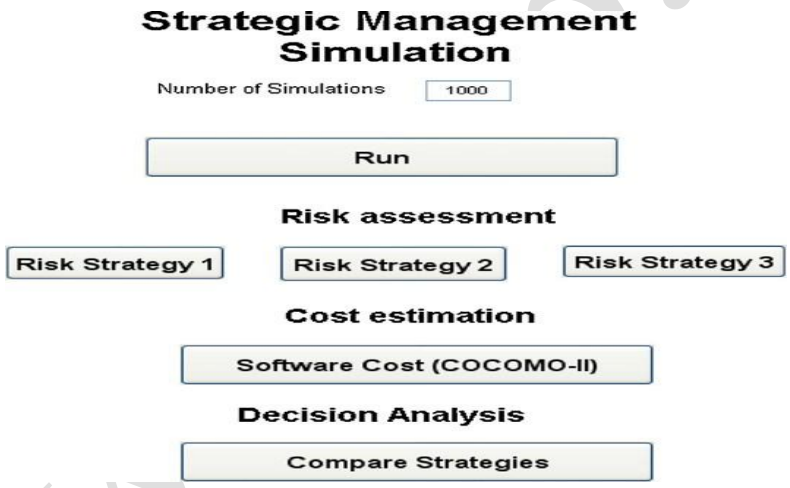

Figure 1: Simulation Framework

Integrated simulation framework identifies the relationship between project development strategies and project management plans and determines the variation and effect of cost, risk, budget and schedule during the different development phases under different strategic decisions. This frame work processing is based on plug and play components, each risk categorisation is based on severity level. The prototype of the proposed simulation model uses the CMMI risk management process model. CMMI divides risk management into three categories: defining a risk management strategy, identifying and analysing risk, and implementing a risk mitigation plan for each identified risk.

III. SOFTWARE MAINTENANCE RISKS MONITORING

To increase the chances of software projects to be errorless, it is important to identify its risks and to monitorSoftware maintenance is a serious complexity in the life-cycle of project systems applications and it is developed for fixing bugs, improving performance or Environment Changing. Software maintenance is categorized into three types: (A) Corrective

maintenance category: Describes about Correction will flexibly upgradable to all requirement. (B). Adaptive maintenance category: Targets on adaptability property will be included for accepting changes with new technologies.(C). Perfective maintenance category: To ensure perfection on different levels of requirement changes and its acceptance will be perfect able to display the output values.

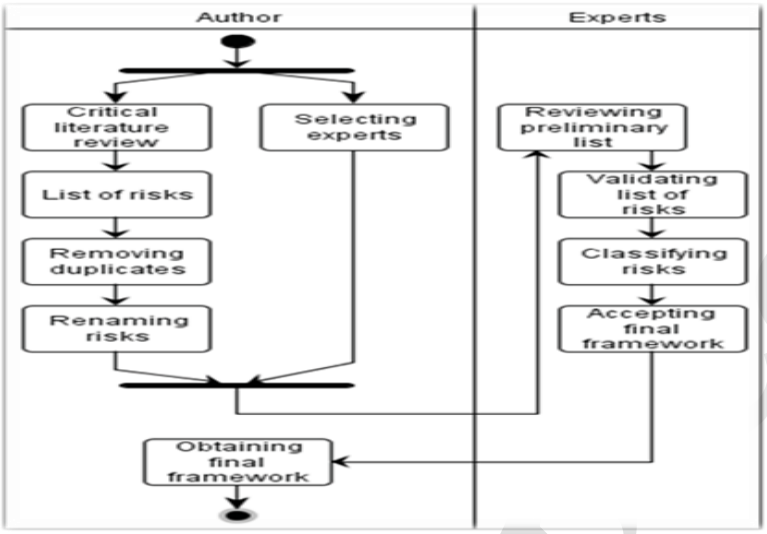

Figure: 2 Activities Diagram of software maintenance risks framework

This Activity chart explores a work flow defines risks affecting in primary level of a software development processing, every stage gives a new approach to author and experts to find some risk resolving techniques without affecting the process of development and maintence activities Speculative Analysis Technique using Awareness Tools (SAT-AT) used the idle information from previous Version files to accurately identify classes of conflicts during design of project. Many software projects include both the bugs of known and unknown nature because of the fact that the total software defects usually exceeds with the available resources. An automation technical method called Genetic-Program [2] was designed for repairing defects which did not had a formal specification, explanation in program. Gentic Program technique does not required formal specifications, program annotations, or special coding practices. GenProg approach is generic, to repair several types of defects. This Genetic program is applied either to the full program source or individual modules. Genetic programming (GP) search for a program variant that retains required functionality but is not vulnerable to the defect.GP introduce three key innovations to address this longstanding risks [3]. **First:** GenProg operates on statement level otherwise called Abstract Syntax Tree (AST).**Second:** based upon hypothesized programs that contains an error in one area partially implements the correct behaviour. **Third:** This type of risks point is critical to Fault localization it is hard and unsolved risks. Genetic program also contains a nice additional features to Risk Repair Census (RRC) like **1. New Repair, 2.Closed-Repairs 3. Repair Quality** by this census Risk reduction will be done with high impactness from the user satisfaction.

IV. BRANCH COVERAGE EXPECTATION

This part focus on complexity measure's aims to provide a technical based support for the tester to find errors in the code. This measure will predict a better way to an automatic test data generator depending upon the program under test. This originality of complexity measure, defines the new measuring model called 'Markov model". This model is used to represent the program to provide an estimation to the number of random test cases that can be generated to obtain a "Concrete Branch Coverage" technique.(BCE) Branch Coverage Expectation experiment's most efficient validation methods said to be Evolutionary and Random Testing techniques, which are the most popular search algorithms for **Automated Test Cases Generator (ATcG)** process. Markov Chain model is used to compute the BCE, Also to estimate the number of random test cases that must be generated to achieve a concrete value of branch coverage. Markov chain is a random sequence of states say 'Xt', where each state depends on the previous one. A state which is reached infinitely often in a finite Markov chain is called positive-recurrence. If every state in Markov chain can be reached from other state, then we say that the Markov chain is irreducible. Markov Chain

model is built from the Control Flow Graph (CFG) of the program, where the states of the Markov chain are the basic blocks of the program. A basic block (BB) is a portion of the code that is executed sequentially with no interruption. Software applications are essential for Industries and software Measurement is a key factor of understanding and controlling software development practices. The requirement's validation is a measure validity check points

- Attribute validity-Attributes and its properties of an entity which possesses.
- Unit validity-A measure used to map an empirical attribute to the formal, mathematical world
- Instrument validity-This instrumentation model defines Data Capturing.
- Protocol validity-Measurement protocols let us measure a specific attribute on a specific entity consistently and repeatedly

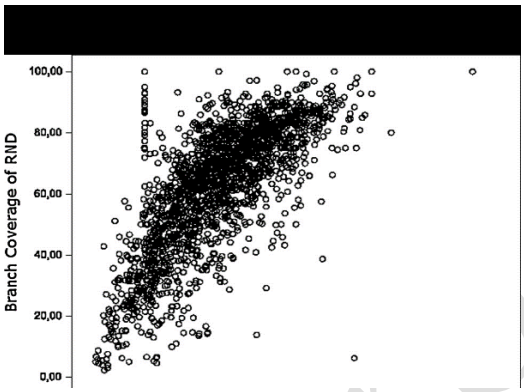

Figure 03: Average Branch Coverage Test Case

Fig. 03shows the obtained average branch coverage with the random test data generator against the BCE measure.Generally Test case generator selects a partial objective (a branch) and uses the optimization algorithm to search for test cases exercising that branch. After the optimization algorithm stops, the main loop starts again and the test case generator selects a different branches. This scheme will repeat until total branch coverage is obtained or a maximum number of consecutive failures of the optimization algorithm. The trend emphasised a clearly gives lower Branch Coverage Expectation values, always produce lowerlevel coverage. We have opened a way to estimate the difficulty to test a program using a'Random Testing Coverage (RNC)'this existing complexity measures or otherwise known as static measures like the nesting degree. The ''Branch Coverage Expectation'' used to provide some knowledge about the difficulty of testing programs. The foundation of this measure is based on a Markov model of the program always provide a theoretically concluded technique.

V. Automated Risk Identification Expert System Management

Experts in risk management recommends that an effective risk identification should be performed by taking into account results of studies done by experts in risk management. Project risk managers should collect documents describing projects characteristics and the corresponding risks identified for them.

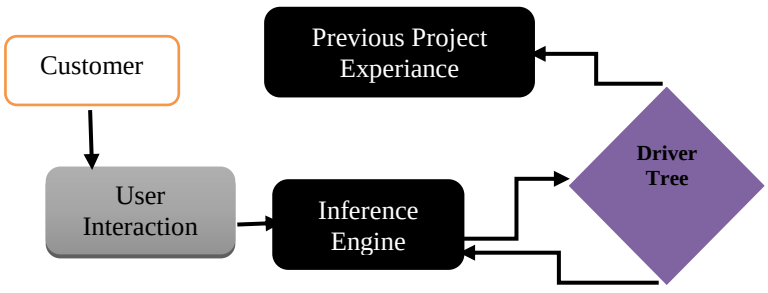

Figure: 04 Expert System Flow

Figure: 04 shows work flow about expert system behaviour based on previous experience learned about Risk identification process. Every Customer interaction will perform some unidentified risks that will interrupt user operational activity, Inference Engine will accept all the request submitted by User's, then starts processing based on Requirements whether it is feasible or not.A *driver tree Or Decision Tree* is a combination of data structurerepresentationwith a set of classification rules, to model inductive learning and decision making abilities. A decision tree learning algorithm is a method to create a model predicts the valuesof output variable and target variable and input variables. Byanalysing the correlations among those values of input and output variables, weusedlearning algorithm to build a *hypothesis.*The efficiency of the driver tree performance can be evaluated using inductive learning algorithm which uses the concepts of *entropy&. It* is a measure based on the occurrence probability of each possible event (*i.e.*, values of the input variables). Expert system must use a list of known risks. With the collection of list can be generated from a collection of documents describes lessons learned in previous projects. Basically, there are two types of risk:

- ***Generic risks***, the most part of projects.
- ***Specific risks:***Deals with threat on specific project under evaluation. Easily detectable by expert system.

VI. FEASIBILITY VALIDATION RISKSUSING RAYLEIGH CURVE

$$\textbf{Effort} = \frac{\textbf{Project size}}{\text{Productivity.Time}} \quad \textbf{B}$$

Above mentioned equation used to calculate a total risk identification process using framework activities. Rayleigh curve used for cost Estimation based calculation with the support of attributes will be given below, this method will extends responsibility of Rayleigh Curve into Risk finding strategy and Maintence Strategy.

Attributes:
Effort- Total Time invested to a framework activity.
Size- Related Project Categories such Small, Big, Large, Very Large project.
Note: **-** Man power will differ depends upon the project Size.
B-Functionality of projects and sizes then Scaling Factor on project.
Productivity- Deals with Organizational Productivity in a period of time.
Time: -Deals with total Schedule of the project.

VII. SAMPLING DATA FRAME RISK DATA

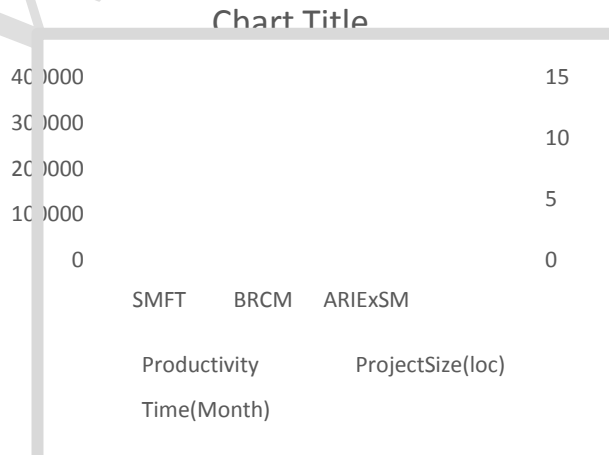

Figure 05 Sampling study on Framework Random Data's

VIII. CONCLUSION

This framework tool discussion we identified different technical shootings to identify Quantitative and Qualitative Risk's which give more defects severity levels and remedies to those unidentified defects. Next proposal on this process is to Framework based design with well freeze requirement's for all kinds of software products. Generally frame works are usually gives a solution for a specified and unexpected issues after some situations like Unit Testing, Integration, Alpha and Beta Testing activities, etc., Among these featured frame work based process Risk's is re-productive process will have different approach in different situations.

References

[1] Masood Uzzafer., "A simulation model for strategic managementprocess of software projects," The Journal of Systems and Software,Elsevier journal, 2013

[2] Claire Le Goues, Thanh Vu Nguyen, Stephanie Forrest, SeniorMember, IEEE, and Westley Weimer, "GenProg: A Generic Methodfor Automatic Software Repair", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 1,JANUARY/FEBRUARY 2012.

[3] Gustafson, A. Ekart, E. Burke, and G. Kendall, "ProblemDifficulty and Code Growth in Genetic Programming," GeneticProgramming and Evolvable Machines, vol. 5, pp. 271-290, Sept. 2004.

[4] Javier Ferrer, Francisco Chicano, Enrique Alba, "Estimating softwaretesting complexity", Information and Software Technology, Elsevier,July 2013.

[5] T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation,Oxford University Press, New York, NY, 1997.

[6] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, Vol. 1, No. 1, 1985, pp. 81-106. doi:10.1007/BF00116251 .

[7] J. Gao and Z. D. Han, "New Decision Tree Algorithm with Restrained Factor Involved," *Physics Procedia*, Vol. 25, 2012, pp. 1871-1878. doi:10.1016/j.phpro.2012.03.324

[8] C. A. R. Morano, C. G. Martins and M. L. R. Ferreira, "Application of Techniques for the dentification of Risk in the E & P Ventures," *Engevista*, Vol. 8, No. 2, 2006, pp. 120-133.

[9] J. Dhlamini, I. Nhamu and A. Kaihepa, "Intelligent Risk Management Tools for Software Development," *Pro-ceedings of the Annual Conference of the Southern Afri-can Computer Lecturers' Association*, Eastern Cape, 2-11 July 2009, pp. 33-40.

[10] A. Elzamly and B. Hussin, "Managing Software Project Risks (PlanningPhase) with Proposed Fuzzy Regression Analysis Techniques with FuzzyConcepts," Int. J. Inf. Comput. Sci., vol. 3, no. 2, pp. 31–40, 2014.

[11] A. Elzamly and B. Hussin, "An Enhancement of Framework SoftwareRisk Management Methodology for Successful Software Development," J.Theor. Appl. Inf. Technol., vol. 62, no. 2, pp. 410–423, 2014.

[12] S. Basha and D. P, "Analysis of Empirical Software Effort Estimation Models," Int. J. Comput. Sci. Inf. Secur., vol. 7, no. 3, pp. 68–77, 2010.

[13] J. Ferrer, F. Chicano, E. Alba, Evolutionary algorithms for the multi-objectivetest data generation problem, Software Practice and Experience 42 (11) (2012)1331–1362.

[14] P.S. May, Test Data Generation: Two Evolutionary Approaches to MutationTesting, Ph.D. Thesis, Computing Laboratory, 2007.

[15] Ashish Sharmaa, Dharmender Singh Kushwaha," Estimation of Software Development Effort from Requirements Based Complexity", Elsevier, C3IT-2012.