ISSN:0975-9646

A Survey on Software Project Planning and Control Using Case Tools

NadanaSundaram P V [#1], Dr.K.Iyakutti [*2]

[#] *Research Scholar,*
Madurai kamaraj University,Madurai, Tamil Nadu, India

[*] *Professor(Rtd),Madurai kamaraj University,*
Madurai, Tamil Nadu, India

Abstract:In software projects, there exists a series of problems in planning and controlling process. In order to avoid the problems, Strategic Management Process (SMP) is designed for identifying and quantifying the risks. The risks are identified by joint monitoring of software planning and project complexity which uses cumulative distribution functions to produce an expected message alert per risk factor. First step for avoiding the risk is planning and controlling, which assists the software engineers by managing not only the quality of software project but also to develop risk-free software products, after identifying the risk. Software project on each module is checked to detect the complexity of the software project at an earlier stage. Evaluating the risk by ranking the software and monitoring the important activities of software project manager is one solution to solve the problems.

CASE tools (Computer-Aided Software Engineering) are a set of application tools and software system methods with the desired end result of high-quality, defect-free, and maintainable software products. CASE tool also refer to development of information systems methods together with automated tools which is used in the software project development process. In CASE tools, computer-based support is used by developers in order to develop and maintain the software projects. CASE tools also enable software engineers to come out from the actual complexities of code when looking at the complex projects. This paper presents software projects planning and control risks. With help of CASE tools the risks are identified, planned and controlled in the software projects. This paper also improves the efficiency and reduces the risk in the system

Keywords*: *Computer Aided Software Engineering, Bayesian Statistics Software Process, Risk Control, Cumulative Distribution Function, Empirical Quantile

1. INTRODUCTION

Software project suffers from many problems like high computational cost, higher delay time in designing the projects, not meeting the actual need of the user, and many systems are being unutilized. These problems are solved using the software risk management which helps the software developer to identify, analyze, and accordingly 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.

Software project development evolves a highly complicated environment in software markets. So, the companies are under huge pressure to structure and to provide a competitive provisioning by reducing project complexity and software project latency time while maintaining software project quality which highly requires the software industries to concentrate more on project complexity. Different strategic management decisions create different sets of risks with different cost commitments. So, each strategic decision requires a project management plan with its own unique budget and schedule of software development.

Computer-Aided Software Engineering (CASE) tool provides programmed support for software development. The aim of CASE tools is to reduce the delay time and computational cost of software development and also to enhance the quality of the systems developed. The main interest in CASE tools is based on potential about increasing productivity, improving quality of the product, improving the maintenance, and making software engineer's task as less hateful and more enjoyable.

Software Project Planning and Control using CASE Tool aims to:

- Handle the risks related to software planning and control dimensions by the software engineers in order to develop risk-free software products.
- To reduce the software project complexity on multiple related project
- Improve the quality of the software risk control and to produce an expected message alert per risk factor

This paper is organized as follows: Section II discusses software projects planning and controlling, Section III shows the study and analysis of the existing planning and controlling methods using CASE tools, Section IV identifies the possible comparison between them and Section V concludes the paper, key areas of research is given as making use of software projects planning and controlling using CASE tools and designing new algorithms and systems for efficient software projects.

2. LITERATURE SURVEY

Strategic Management Process (SMP) [1], simulated the impact of planned types of decisions namely, cost, risk, budget requirement and scheduling of the project using simulation model and 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 which also provides a critical insight that efficiently identified the best strategy. But, while analyzing SMP, it showed that planning and control had to be extended with cost effective software engineering tools for meeting out the resources required for a project.

Speculative Analysis Technique using Awareness Tools (SAT-AT) [1] used the idle information from previous version files to accurately identify classes of conflicts during design of project. But, SAT-AT did not provided solution for qualitative and quantitative result related to risk during the planning and control stage on software projects.

One of the major problems in software project is the quality of the software. 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 automated method called GenProg [3] was designed for repairing defects which did not had a formal specification, explanation in program. GenProg applied an extensive method of genetic programming to design a method in order to retain the functionality of project. Though repairs generated were examined in a quantitative and qualitative manner, automatic repair remained unaddressed.

Branch Coverage Expectation (BCE) [12] investigated complexity measures related to program by taking into consideration the most important features of a program. Markov model was used in BCE estimates the test cases required to reach the coverage level which resulted in the increase of correlation while testing a program by reducing the software testing complexity. Yet, the software testing complexity was not identified in the early stage. One of the major problems associated to the software industry is software project failure.

Ambient Intelligent applications [2] provided a middleware architecture designed specifically for safety critical applications which provides the developer with different services during the runtime verification. The architecture monitored and measured the correctness of the project continuously using a visual tool providing automatic generation of setting up parameters by improving the runtime. But, it was not applied for wide range of applications.

Control Objectives for Information and related Technology (COBIT) [10], briefed a framework which helped the organizations to effectively manage and eventually analyze the suppliers in a multi sourced environment. COBIT used key performance indicators (KPIs) to perform quantitative analysis. Though quantitative analysis was included, but was limited to seven business entities.

An Integrated Framework for Risk Response Planning (IF-RRP) [8] was planned to provide mechanisms during project management, for minimizing and correcting the errors, using an efficient matrix representation called structure matrix where Genetic algorithm was also introduced in IF-RRP for large projects. Though risk propagation behavior was evaluated using sequential forward selection greedy algorithm and genetic algorithm, but risk was not measured involving multiple related projects.

To minimize the risk factors, a new technique with respect to time was analyzed by applying the incline construction principles [11] using last planner system. To minimize the project complexity with respect to software project completion time, Percent Expected Time-overrun (PET) and Percent Plan Completed (PPC) was used. Still, very simple and highly efficient, the method was highly complex with respect to cost.

Speculative Analysis Technique using Awareness Tools (SAT-AT) [9] identified different types of conflicts and risks using Crystal, which helps developers to identify, manage and prevent conflicts. Though, computational cost was reduced using SAT-AT, qualitative and quantitative analysis was not provided. In [13], a new type of log-linear regression model was designed on the basis of use case point model (UCP), to measure the software project using use case diagrams to measure the software project failure in the early stage.

Speculative Analysis (SA) Technique [14] was designed; which was efficiently used previously-unused information from different version control methods to measure accurately and diagnose different conflict classes. But, SA technique was not full proof towards qualitative and quantitative analysis. Stepwise Regression Analysis (SRA) Techniques Bayesian Statistics [15] introduced a novel statistical model which served as a risk management model using quantitative results. Herewith, the method was proven to be efficient to decrease software risk. Though, the size of dataset was incomplete.

To increase the success rate of software project, risk management is highly significant in software engineering in which the design of expert system was constructed in [16] for efficient risk identification with the aid of lessons obtained from several unique and similar projects which is designed by software programmers. With help of this, the bug that may occur in the past was reduced considerably at a significant rate. Though efficient, it identified the risks based on the previous projects but not error level to new projects. Localization of faults in web applications was designed in [18] using conditional and functional call statements. However, qualitative and quantitative factors were not examined.

3. Techniques Of Software Projects Planning And Control

Project planning and control are the two important operations to ensure project efficiency and the project effectiveness. Software engineering is a process methodology for development and application of information processing systems. If it is applied constantly, this method helps to avoid number of errors in system development, particularly in software development. Project planning and controlling is the part of project management which is essential to the way of proceeding and also describes the coherent methodology for project management, in which development and maintenance costs are reduced.

3.1 DETECTION OF CONFLICTS AND RISKS

Conflicts among developers arise in collaborative development which slows the progress and also decreases the quality. Conflicts can be in any of the form either textual or higher-order. A textual conflict arises when two developers make inconsistent changes to the same part of the source code. To prevent subsequent changes from overwriting previous ones, a version control system (VCS) allows the first developer to publish changes, but prevents the second developer from publishing until the conflict is resolved automatically or manually.

Higher order conflicts arise when there are no textual conflicts among developer’s changes, but those changes are semantically incompatible. Higher-order conflicts can cause compilation errors, test failures, or other problems, and are problematic to detect and resolve in practice

3.2 STRATEGIC MANAGEMENT PROCESS

Strategic management process simulation model is an integrated framework which maps strategic decisions with the cost estimation, risk management and project management planning. The risk management identifies and quantifies the risk while the cost estimation produces the random estimated cost in man-months units and also integrates it with the risk. The project management transforms the cost into the budget and the schedule of software projects.

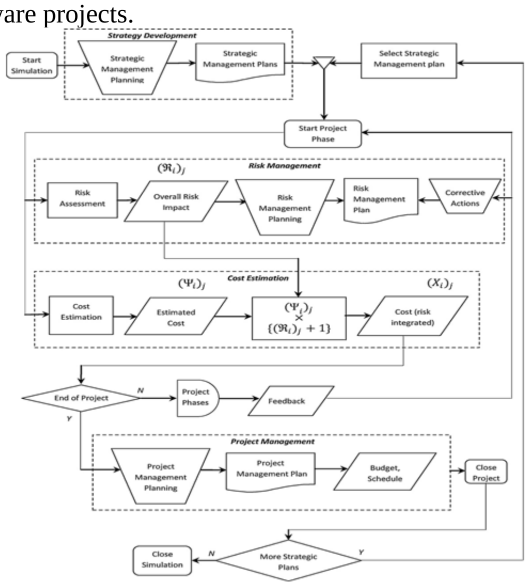

Fig. 3.1 Simulation model for strategic management process of software projects

A simulation model for the strategic management process of software development projects model simulates the inferences of strategic decisions on the factors such as cost, risk, budget and schedule of software projects. The simulation of the project management planning determines the budget and schedule required parameters for a project. Different strategic management decisions consists different sets of risks, which requires different cost commitments. So, each strategic decision requires a project management plan with its own unique budget and schedules of software development. As a result, the simulation model approximates the risk and cost under different strategic decisions and maps them according to the project management plans. Hence, for the development and management of software projects, the integrated framework helps to identify the best strategic option.

The simulation model is unclear because it contains generic plug and play components which facilitate the use of any set of risk assessment, cost estimation models and project management tools. For that reason, it provides a flexible solution to software organizations and managers of software development projects. The simulation model is applied to a case study, which showed the effect of different strategic decisions on the risk and cost of the different phases of software development and finally on the budget and schedule required to complete the project. It therefore provides critical insights in identifying the best strategy for the development of software projects.

Fig. 3.2 Strategic management applications

3.3 MONITORING SOFTWARE MAINTENANCE PROJECT RISKS

Software maintenance projects consist of certain feature which makes them different from other engineering ones and includes increased complexity and higher project failure rates. In order to increase the chances of software projects to be errorless, it is necessary to identify its risks and to monitor them.

Software maintenance is a serious issue in the life-cycle of project systems applications and it is developed for fixing bugs, improving performance or other attributes or adapting them within a changing environment.

Software maintenance is categorized into three types:

1. Corrective maintenance category
2. Adaptive maintenance category.
3. Perfective maintenance category

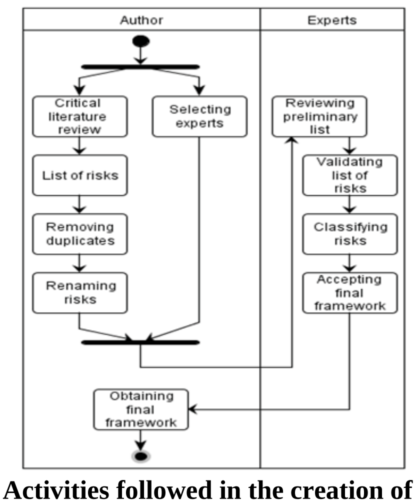

Fig. 3.3 Activities followed in the creation of software maintenance risks framework

Though, success in software maintenance projects is not guaranteed and failures in software maintenance also played a major role in many high-profile disasters. In some cases, these failures directs to undesired problems. The system will not perform well, if the adopter firm fails to perform software maintenance and the project systems fails to fulfill initial expectations. Besides, daily business activities may be delayed which leads to bad impacts on company stability. As a result, software maintenance is critical to the success of enterprise systems adoption.

3.4 INTEGRATED FRAMEWORK FOR RISK RESPONSE PLANNING

Engineering project managers face a challenge to allocate tight resources for managing interdependent risks. In the risk response planning, a design structure matrix representation is used to capture risk interactions and build a risk propagation model for predicting the global improvement effects of risk response actions. For representation, a genetic algorithm is used as tool for choosing response actions and allocating budget reserves. Comparison with a Sequential Forward Selection greedy algorithm shows the superiority of the genetic algorithm search for optimal solutions and its flexibility for balancing mitigation effects and required budget.

An integrated framework for risk response planning consists of five-step framework. They are:

1) Building project risk network;
2) Defining objective function;
3) Identifying budget constraints;
4) Identifying potential response actions;
5) Optimizing risk response plan.

Through modeling risk interactions, the framework makes it possible to analyze risk propagation behavior and thus to anticipate the overall effects of response actions on the global risk network and it can also guide the project manager design some non-conventional actions on risk interactions which mitigates risk propagation instead of risk occurrence.

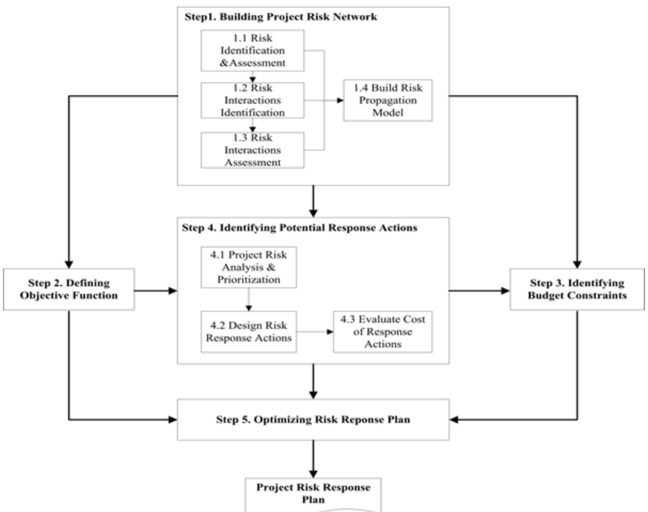

Fig. 3.4 Framework for risk response planning

3.5 STEPWISE REGRESSION ANALYSIS TECHNIQUES

The stepwise multiple regression analysis techniques reduces software maintenance risks in a software projects. However, these statistical measures will be performed using stepwise multiple regression analysis in order to compare the risk management techniques to each of the software maintenance risk factors to identify if they are effective in reducing the occurrence of each software maintenance risk factor and selecting the best model. The achievement of software project risk management will improve the probability of software project success. In addition, historical data from database cannot be obtained until using some techniques.

The main goal of this technique is to mitigate software maintenance risks in software project management based on quantitative. After identifying the software maintenance risk factors and risk management techniques of software projects in the software development organizations, ranking the software risk factors and risk management techniques is done according to their importance, severity and occurrence frequency, to identify the activities performed by software project managers to manage the software maintenance project risks which are identified by using stepwise regression analysis modeling.

Table 3.1 Illustrate Top 10 Software Maintenance Risk Factors in Software Project Lifecycle Based on Researchers

Phase	No	Software risk factors	Frequency
Maintenance	1	Inadequate knowledge/skills (Addison and Vallabh 2002; Addison 2003; Aloini et al. 2007; Aritua et al. 2011; Cliff Mitchell 2011; Keil et al. 2002; Lyons and Skitmore 2004; Schmidt et al. 2001; Sumner 2000), all remaining for Schmidt in various cases.	11
	2	Inadequate change management (Aloini et al. 2007; Keil et al. 2002; Lyons and Skitmore 2004; Nakatsu and Iacovou 2009; Schmidt et al. 2001; Sumner 2000)	6
	3	Corporate politics with negative effect on software project (Addison 2003; Aritua et al. 2011; Chen and Weng 2009; Han and Huang 2007; Schmidt et al. 2001)	5
	4	Lack of resources and reference facilities (Aritua et al. 2011; CHAOS 1995; Han and Huang 2007; Lyons and Skitmore 2004)	4
	5	Lack of top management commitment and support and involvement (Aloini et al. 2007; Keil et al. 2002; Schmidt et al. 2001; Sumner 2000).	4
	6	Shortfalls in externally furnished components, COTS (Boehm 1991, 2002b, 2007)	3
	7	Legacy software project (Boehm 2002a)	1
	8	Acquisition and contracting process mismatches (Boehm 2007)	1
	9	User documentation missing or incomplete (Chen and Huang 2009)	1
	10	Harmful competitive actions (Khanfar et al. 2008).	1
		Total frequency	37

4. COMPARISON OF SOFTWARE PROJECT PLANNING AND CONTROL TECHNIQUES

In order to compare the feasibility ratio of software project planning and control techniques, test cases are taken to perform the experiment. Feasibility ratio is defined as the ratio of speed required to complete the rework γ and the sum of speed to perform the rework with rework rate x time the new software productivity rate β which is given as

$$FR = \frac{\gamma}{\gamma + x\beta}$$

The comparison takes place on existing software planning and control techniques using CASE tools such as Strategic

Management Process (SMP), Integrated Framework for Risk Response Planning (IF-RRP), Speculative Analysis Technique using Awareness Tools (SAT-AT), and Stepwise Regression Analysis (SRA).

4.1 FEASIBILITY RATIO:

Table 4.1 Tabulation for Feasibility Ratio on different software projects planning and control techniques

TEST CASES	Software feasibility ratio (%)			
	SMP	SAT-AT	IF-RRP	SRA
1	38	35	45	42
2	43	38	53	48
3	48	42	58	52
4	52	45	50	49
5	47	40	63	48
6	55	53	65	56

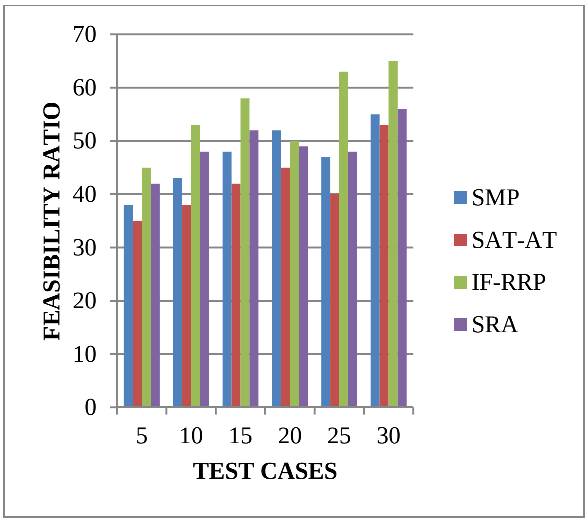

Fig.4.1 Feasibility Ratio on different software projects planning and control techniques

Fig 4.1 describes the feasibility ratio on various numbers of test cases. The feasibility ratio is measured in terms of percentage (%). As the number of test case increases, feasibility ratio is increased in the integrated framework for risk response planning (IR-RRP). The experiment shows that IF-RRP increases the feasibility ratio when compared with the SMP, SAT-AT, and SRA. It can be seen that the integrated framework for risk response planning shows great advantage.

Software projects planning and control techniques of IF-RRP has 20 – 25 % higher feasibility ratio when compared with the SMP and 25-30% higher feasible when compared with the SAT-AT. Planning and control techniques in IF-RRP is approximately 5% higher feasible when compared with the SRA.

4.2 DELAY TIME:

Delay time defines the amount of time gets delayed during the identification of risk which is measured in terms of milliseconds (ms). The below table (Table 4.2) describes the delay time on different projects planning techniques such as SMP, SAT-AT, IF-RRP and SRA.

Table 4.2 Tabulation for Delay Time on different software projects planning and control techniques

TEST CASES	Delay time (ms)			
	SMP	SAT-AT	IF-RRP	SRA
1	75	78	77	80
2	68	75	73	77
3	64	71	69	73
4	65	68	66	70
5	68	72	70	74
6	60	65	64	69
7	52	60	58	65

The delay time of SMP is comparatively lesser compared to all other existing software projects planning and control techniques. The delay time of SMP is 5-10 % lesser when compared to SAT-AT technique and 3-5 % lesser when compared to IF-RRP technique. Finally, the SMP planning and control technique is 7-10 % lesser when compared to SRA method.

When the delay time gets decreased the efficiency and effectiveness gets automatically increased. Delay time plays vital role in increasing the efficiency of the methods.

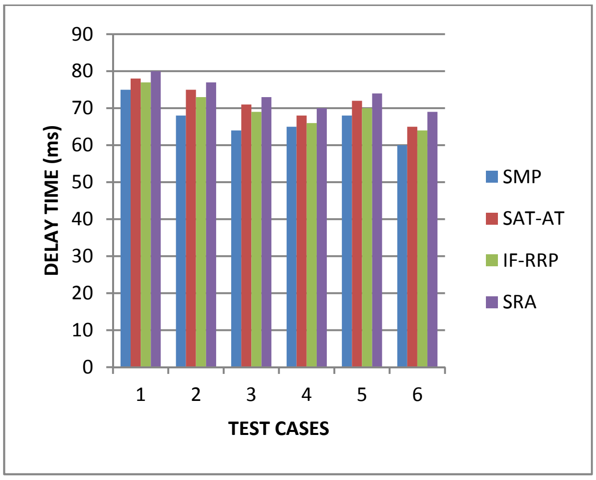

Fig.4.2 Delay Time on different software projects planning and control techniques

5. CONCLUSION

Discussion about existing software project planning and control techniques using CASE tools such as, Strategic Management Process (SMP), Integrated Framework for Risk Response Planning (IF-RRP), Speculative Analysis Technique using Awareness Tools (SAT-AT), and Stepwise Regression Analysis (SRA). Strategic Management Process identifies the risk and quantified the risk with cost assessment and after analyzing SMP, it revealed that planning and control needs to be extended with effective software engineering tools for attaining the required resources.

Existing Speculative Analysis Technique using Awareness Tools provides particular information and accurately identifies important classes of risk occurrence between collaborative team members. SAT-AT also analyzed the important types of conflicts and risks in the early stage using a tool, Crystal between collaborating team members and classified the risks for maintaining software.

Existing Integrated Framework for Risk Response Planning was developed for providing support in decision making during project response risk planning and also used sequential forward selection greedy algorithm and genetic algorithm which analyzed the risk propagation behavior. Stepwise Regression Analysis serves as a risk management model using quantitative results and proven to be efficient to reduce software risk. After analyzing the review shows that software projects planning and control methods are becoming efficient using CASE tools.

Observation was that the building of multiple related projects on existing techniques also reduces the complexity of the projects. This helps in decreasing the delay time which helps in improving the efficiency. The wide range of experiments evaluates the comparative performance of the various planning and control techniques in software projects using CASE tools. Finally, the result shows that the planning and control technique of software projects performs constantly over a wide range of experimental parameters.

One of the limitations of existing software planning and control management is to minimize the project complexity on multiple related projects, to achieve higher ranking effectiveness and to improve the level of feasibility ratio with relatively lesser amount of latency time. But it is not achieved in the existing techniques. Another limitation of existing methods does not offer qualitative and quantitative result for planning and control risks on software projects. In addition more planning and control methods are developed but the efficiency and accuracy are not improved due to higher delay time. Survey has evaluated the planning and controlling efficiently using CASE tools for improving the efficiency and accuracy.

The future direction on software projects planning and control techniques using CASE tools can be taken as a way forward to develop a risk-free software products to handle the risks related software planning and control. In addition, planning aspects need to reduce the software project complexity on multiple related functions. Further improvement should be proceeded in the quality of the software risk control for producing an expected message alert per risk.

REFERENCES

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

[2] Antonio Coronato, and Giuseppe De Pietro, "Tools for the Rapid Prototyping of Provably Correct Ambient Intelligence Applications", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 4, JULY/AUGUST 2012

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

[4] Sanjay Murthi, "Preventive Risk Management for Software Projects", IEEE Software Development, Sep 2002

[5] Liu Jun, Wang Qiuzhen, Ma Qingguo, "The effects of project uncertainty and risk management on IS development project performance: A vendor perspective", International Journal of Project Management, Elsevier, Nov 2011

[6] Nizam Uddin Ahameda, Kenneth Sundaraja, R. Badlishah Ahmadb, Matiur Rahmanc, Asraf Alib, "A framework for the development of measurement and quality assurance in software-based medical rehabilitation systems", International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Elsevier

[7] Lazaros Sarigiannidis, Prodromos D. Chatzoglou, "Software Development Project Risk Management: A New Conceptual Framework", Journal of Software Engineering and Applications, 2011

[8] Chao Fang., Marle, F., Min Xie; Zio, E., "An Integrated Framework for Risk Response Planning Under Resource Constraints in Large Engineering Projects," Engineering Management, IEEE Transactions on (Volume: 60, Issue: 3), 2013

[9] Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin., "Early Detection of Collaboration Conflicts and Risks," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 39, NO. 10, OCTOBER 2013

[10] Thomas Ph. Herz, Florian Hamel, Falk Uebernickel, Walter Brenner, "Toward a model of effective monitoring of IT application development and maintenance suppliers in multi sourced environments", International Journal of Accounting Information Systems, Elsevier, Jan 2013

[11] Usama Hamed Issa, "Implementation of lean construction techniques for minimizing the risks effect on project construction time", Alexandria Engineering Journal, Elsevier, July 2013

[12] Javier Ferrer, Francisco Chicano, Enrique Alba, "Estimating software testing complexity", Information and Software Technology, Elsevier, July 2013

[13] Ali Bou Nassifa, Danny Hob, Luiz Fernando Capretza, "Towards an early software estimation using log-linear regression and a multilayer perception model", The Journal of Systems and Software, Elsevier, Aug 2012

[14] Yuriy Brun., Reid Holmes, Michael D. Ernst, and David Notkin., Early Detection of Collaboration Conflicts and Risks," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2013

[15] Abdelrafe Elzamly., Burairah Hussin., "Mitigating Software Maintenance Project Risks with Stepwise Regression Analysis Techniques," Journal of Modern Mathematics Frontier Volume 3 Issue 2, doi: 10.14355/jmmf.2014.0302.02, June 2014

[16] Julia Botan Machado, Silvio do Lago Pereira, "Automatic Risk Identification in Software Projects: An Approach Based on Inductive Learning", Intelligent Information Management, Oct 2012

[17] Noela Jemutai Kipyegen, Waweru Mwangi and Stephen Kimani, "Risk Management Adoption Framework for Software Projects: A Case Study for Kenyan Software Project Managers and Developers", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012

[18] Shay Artzi, Julian Dolby, Frank Tip and Marco Pistoia, "Fault Localization for Dynamic Web Applications", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 2, MARCH/APRIL 2012

[19] Ayse Tosun Misirli, Member, IEEE, and Ayse Basar Bener, "Bayesian Networks For Evidence-Based Decision-Making in Software Engineering", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 40, NO. 6, JUNE 2014

[20] Olabiyisi S.O.1, Adetunji A.B2, Olusi T.R3, "Using Software Requirement Specification as Complexity Metric for Multi-Paradigm Programming Languages", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3, March 2013)