Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 1– No.3, February 2015 – www.caeaccess.org

Generic Approach for Goal driven Software Requirement Risk Management

Shruti Patil
Dr.D.Y.Patil School of Engineering &Technology,
Savitribai Phule Pune University, Pune,
Maharashtra, India

Roshani Ade
Dr.D.Y.Patil School of Engineering &Technology,
Savitribai Phule Pune University, Pune,
Maharashtra, India

ABSTRACT
An analysis of many software program assignments in between 2011, as well as 2014, shows an interesting pattern. Author identified areas which may lead to project failure which are weak requirement analysis and management, weak cost calculating, weak handling of requirement change requests, weak milestone monitoring and requirement gold plating habit. By comparison, prosperous local software program assignments tended to be much better than onsite software development and management. Maybe the most interesting part of most of these many problem areas can be that each is regarding project managing instead of using technical personnel. We focused over the impact of software requirement change requests and requirement gold plating while dealing with onsite project assignments. Author also evaluated new model to avoid global software engineering requirement failure, which in turn curtails the estimated time and budget with client satisfaction. In this paper, Author discuss the ideas that support requirements reflection as a means to articulate some of the outstanding research challenges.

General Terms
Software Engineering

Keywords
Risk analysis, Requirement change risk , software risk, Global Software Development

1. INTRODUCTION
Requirements are tactful to the context in which the system-to-be must operate. Where such context is well-understood and is inert or evolves pokily, subsisting RE approaches can be brought to exercise beneficially. Additionally, furthermore, development approaches are abiding confronted to develop systems to execute in contexts that are capricious over small periods in ways that are deficiently affirmed.

Such systems lack to be able to deviate to new contexts dynamically, but the contextual deniability that desires this self-adaptive adeptness makes it rigid to construct, adjure and control their requirements. Aberrant contexts may desire asymmetric requirements trade-offs. Sudden contexts may even direct to comprehensive alpha requirements. To workout with such uncertainty, approximations should be run-time entities that can be conceptualized over in order to accept the degree to which they are existence engorged as well as to assist alteration determinations that can take benefit of the systems' self-adaptive delegation. Author take our apotheosis from the reality that accurate, absent approximations of software edifices applied to be assessed design-time-only entities but approximate consideration emerged that constitutional activities could be circumscribed at run-time too, assisting systems to reconfigure themselves according to adjusting context. Author chooses to apply comparative approaches to abduct requirements consideration.

At the core of conforming requirements engineering (RE) is the thirst to comprehend the dilemma in order to construct the requirements prototype, enclosing approaches, domain appropriations as well as requirements. Fundamental in this is the appropriation that the surrounding context is conceivably idle and can be a affirmed adequately well to affirm the requirements prototype for a feasible breakthrough to be developed with confidence. In the conduct, surrounding contexts are seldom idle over long courses, and their sheer degree sometimes bans capitulation. Furthermore, RE assists many of approaches capable of alleviating or bypassing these dilemmas assisted alteration occurs crawling enough to deal developers to approximate the allegations and abduct proper exercise.

Incrementally, although, considerations are being authenticated for problem contexts that are subject to adjust over fewer periods as well as in manners that are deficiently affirmed. In short, this is due to the delegation of self-adaptation has adjusted, ascribing aims for systems to acknowledge at run-time to altering context. For example, adaptive middleware approximations assign software elements assisting asymmetric advantageousness or degree of assistance to be assigned at run-time. Analogous architectural adaptively [1] has made it technically and competitively attainable to effect approximations, such as smart routers [2] that are accomplished to optimize their behavior to controlling circumstances such as network loads. Complementing the bottom-up driver for self-adaptive systems assigned by altered software technology, is a problem-driven impetus engaged by a many coincidentally essential real-world dilemmas alike as disaster evaluating as well as smart enterprise control. The average alternate in each of these dilemma domains is the degree for rapidly-changing, hard-to-understand surrounding contexts [3].

2. LITERATURE REVIEW
Risk management was commenced into software project management by Whitmore [4] and Hibshi et.al. [5]. Software Project Risk Management is an array of directs or conducts, which can determine, anatomize, as well as monitor the risk causes and accumulate the accomplishment approximate of the project. Software Project Risk Management could control estimate, schedule, of the project, etc. [6]. Primary step is risk consideration which confounds risk detection, risk condensation, and risk prioritization.

Risk detection needs balanced detection as well as taxonomy of the risk drives. Risk hypothesis approximates authenticate of each discovered risk cause and determines the connections among risk bases, as well as between risk bases and project

consequence. Risk prioritization chooses the anticipation array in coordinating each risk element [7].

Next step is risk coordination that confounds risk approximating and risk monitoring. Risk approximating comprises not only approximating for each risk element, but also controlling the detached concludes with each other [8]. Consecutive monitoring of the steps of risk elements, analysis of the convenience of the risk-control landscape, and the quick apotheosis of converging risks are needed during and after the facilitation of the consideration. Currently, a communicative count of inline analyzes on risk determination and approximating in the field of software project risk management has become feasible. This paper intersects on 2011-2014 data based risk analysis, on forecasting the likelihood of benefit for software development projects; hence, author exclude analyzes that are based on the core argument or expert judgment, which are approximations often applied in project risk assignment [9].Author also analyzes that anatomize software dependability, cost, security, etc. To the best of our information, there is no definite study on the topic of requirement risk management for agile software project. Risk benchmark is more broadly examined, and the prototyping approaches mainly constitute the application of algorithmic benchmark. This analysis is based on approximate approaches. For example, Hu et al. [10] applied constitutional equation prototyping to develop an exploratory prototype for benchmarking as well as approximating the associations between software project risks and project endeavor. Hibshi et al. [5] approximated many precipitous risk elements and their consequences on software project gain applying regression benchmark. Their analysis determined that the occurrence of approached contributor and the degree of confidence that the clients and employers have in the project manager and the development team are the most authoritarian components for project advantage.

Hu and Yong [11] demonstrated a prototype applying directing elemental benchmark based on a survey of 50 project managers, catechizing the associations between IS achievement allocations and risk elements. In summary, these analyzes attempt to determine the accomplished information about risks rather than forecast the overall risk degree of an ongoing project. There are relative few studies available on risk planning from 2011-2014 literature as follows: Examines that choose approximate approaches to acknowledge the convenience of risk-control activities. For example, Heidrich and Jens [12] applied logistical regression benchmark to adjure the validity of the risk-option mappings of an IT expenditure option-based risk management circumference. Wautelet et. al. [13] chose Pearson's association to examine the association between risk-reduction conducts and risk elements. Hsu and Wen-Ko [14] determined that client partnering is definitively applicable to higher client assist, under remaining risk, as well as beneficial project conduct. He and Yong-xiu et al. [15] ascertained the existence of an authoritarian disproving association between standardization as well as remaining performance risk. Heidrich et al. [12] admitted and contrasted the exercises of four risk reduction conducts on risk elements based on 50 software projects.

Asnar et al. [16] acquired evidence from 86 project managers from the Project Management Institute (PMI) also acknowledged that the level of command behaviors during the system conception approach has a definitive exercise on software adaptability. In summary, approximate approaches are often addressed to analyze the validity of risk-control conducts on risk elements.

Other literature analyzes that heading to drive the optimal risk-control activity set applying cipher approximating technology. For example, Eric et. al. [17] developed a risk response prototype, which can approximate the associating conducts of multiple risk curtailment activities, as well as the impacts of accessory risk consequences. This risk-response prototype can also approximate the total risk breakthrough dependent on complex composites of risk decrement activities.

The prototype conducts integer-programming technology to drive the most cost-effective inclusion of risk detraction activities. Zwikael et. al.[18] demonstrated an economic optimization prototype for determining risk derogation behaviors in the risk counter-reply approximating phase of CMMI-based project. Their prototype appraises the appropriate barriers that bound the construction of detraction conducts, and can be determined applying integer programming technology.

The Standish CHAOS Report, which studied 9,236 IT projects, discover the main 3 reasons of project failure: lack of user input, incomplete requirements and changing requirements [19]. As per the survey, factors affecting project performance are shown in Figure 1.

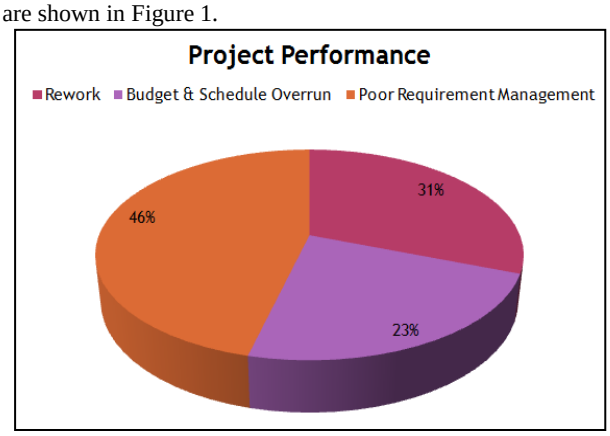

Fig 1: Factors affecting project performance

3. INTEGRATIVE FRAMEWORK FOR COGENT SOFTWARE PROJECT RISK APPROXIMATING

To accomplish an integrative risk assignment that can ascribe determination countermeasure from risk benchmark to blueprinting, Author choose an integrative framework for cogent software project risk blueprinting (CSPRB). Our configuration is constituted of three key elements:

- Risk Database
- Risk benchmark
- Risk blueprinting

3.1 Elements of the proposed configuration

Project Risk Database is an assembly of risk elements as well as final consequences of existing software projects. It ascribes project samples for the risk benchmark module. The sample size, as well as degree, definitively control the validity and dependability of the risk benchmark module and previously, of the risk blueprinting module. Risk benchmark Module acquires a risk benchmark prototype as its focus to benchmark and forecast the project speculates. The prototype accumulates the grades of risk elements as the input and acknowledgments the forecasted approach consequences as output. Risk approximating Module accumulates the risk-control activities

list, the many to- many associations between activities and risk elements, and the development cost of activities as input. It outputs a cost-minimal risk control activity set that can assist in accomplishing that the forecast of the project consequence is achievement. The module conducts through generate and test based mechanism as follows:

- A new user activity category is driven according to the list of risk-control activities, the development costs of the activities, and the many-to-many association between risk control activities and risk elements.
- The consequence of the user activity attribute on the footings of risk elements is approximated. Allocated the alpha marks, the risk benchmark module forecasts the project result. If benefit is forecasted, the balance development expense of the user activity caste is contrasted with that of the present appropriate set. The minimum expensive one is chosen as the alpha optimal apportion.
- If adhering traversed the total search dimension, the module outputs the present optimal apportion, which is a cost minimal activity apportion. Otherwise, it conducts to the alpha stage.

3.2 Stage of approaching the proposed configuration

The initial three stages keep eye on developing an advantageous tool/model for integrative risk benchmark as well as blueprinting whereas the last couple of stages is to address the facility to an ongoing software project.

Stage 1: Information assortment

Initially, the risk elements that will be applied in the risk analysis module are determined. Further, project samples are accumulated according to the detail of risk elements to compose the risk database. Accumulating good software project samples is a long-term as well as invaluable approach.

Stage 2: Risk benchmark module developments

Now, a correct prototyping approach for risk benchmark is determined by contemplating the differences of the acquired project samples as well as the interpretability of the constructivist prototype. Further, a risk benchmark prototype is developed based on the information ascribed by the risk database.

Stage 3: Risk module developments

Initially, the many-to-many association prototype between risk controls activities as well as risk elements is discovered. The prototype is to bar the extent of the risk blueprinting dilemma considered in the present analyze.

Stage 4: Risk benchmark

Initially, the present betokens of the risk elements of the project are approximated. Further, the risk benchmark module is exercised to forecast the project consequence. If the forecasted consequence is a defeat, the ensuing risk approximating is resulted. The approaches of risk benchmark should be frequently conducted.

Stage 5: Risk blueprinting

Initially, the parameters of risk blueprinting, comprising a set of user risk-control activities, the many-to-many association between activities as well as risk elements and the development amounts of activities are approximated. Furthermore, acquiring the above parameters as input, the risk blueprinting module outputs the cost-minimal activity apportion.

Finally, the project stakeholders approximate the caused activity apportions as well as alters it according to real-life circumstances. Consequently, the risk consideration mechanism is begun to conduct the driven conception.

4. GOAL DRIVEN APPROACH FOR REQUIREMENT RISK MANAGEMENT

There are several risks in the software engineering which is not easy or impracticable to recognize all of them. A few most notable risks in software project are classified as software requirement risks, software scheduling risk, software cost risks, and software quality risks. The software requirement risks are listed as below

1. Lack of analysis of requirements
2. Wrong requirements
3. Misunderstood requirements
4. Poor explanation of requirements
5. Requirements ambiguity
6. Lack of requirements management skills
7. Changing requirements
8. Inadequate requirements
9. Impossible requirements
10. Invalid requirements

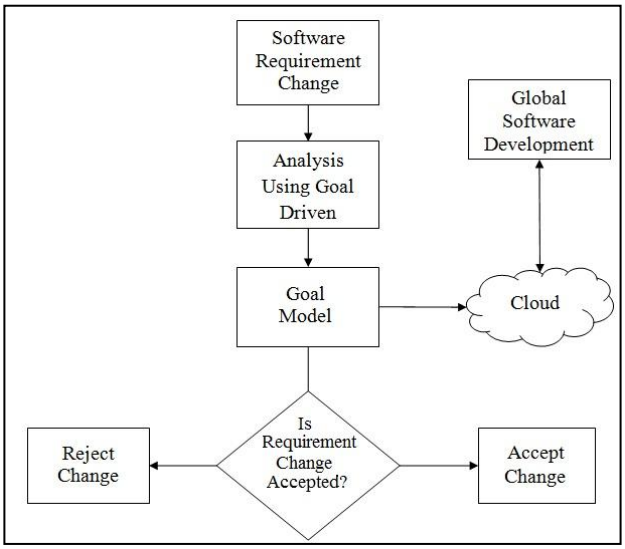

Fig 2: Requirement change management using goal driven approach

The requirement management is one of the major challenges and it causes several concerns when many teams are dispersed over globe and attempt to meet the client's expectation. Figure 2 shows the goal driven approach for requirement change management over global software development (GSD). In this framework cloud is used to share goal model over GSD[20-23].

When Author consider a GSD environment there are many people included so Author use cloud which offers a clear communication, for software engineers working at various

locations to develop software. It is essential to form a shared storage area which is accessible to all team members dispersed globally.

5. CONCLUSION

Risk benchmark and blueprinting are complicated, eventuating in complexities to control risks authoritatively through individual decision. So far, the existing risk analysis area needs the benchmarked integrative cogent prototype for risk benchmark and blueprinting. The proposed CSPRB is the initial integrative configuration for cogent SPRM, which approaches at developing a cost-minimal risk-control activity caste. In the accrual, the benchmarked prototype will be accepted based on real software project information.

In future author will carry out more case studies of large scale organizations at requirement change risk management procedure for advance assessment.

6. ACKNOWLEDGMENTS

The authors would like to thank the support throughout the research from different researchers and colleagues.

7. REFERENCES

[1] Younis, Awad A., Yashwant K. Malaiya, and Indrajit Ray. 2014. Using Attack Surface Entry Points and Reachability Analysis to Assess the Risk of Software Vulnerability Exploitability. IEEE 15th International Symposium on High-Assurance Systems Engineering (HASE).

[2] Islam, Shareeful, Haralambos Mouratidis, and Edgar R. Weippl. 2014. An empirical study on the implementation and evaluation of a goal-driven software development risk management model. Information and Software Technology.

[3] Babar, Muhammad Ali, and Christian Lescher. 2014. Editorial: Global software engineering: Identifying challenges is important and providing solutions is even better. Information and Software Technology

[4] Whitmore, 2014. Threat analysis in the software development lifecycle. IBM Journal of Research and Development

[5] Hibshi, Hanan, 2014. Rethinking Security Requirements in RE Research. Tech. Rep. Report CS-TR-2014-001, Univ. Texas at San Antonio..

[6] Yang, Ming 2014. "The Measurement and Analysis of Software Engineering Risk Based on Information Entropy." Proceedings of International Conference on Soft Computing Techniques and Engineering Application. Springer India.

[7] Creemers, Stefan, Erik Demeulemeester, and Stijn Van de Vonder. 2014. A new approach for quantitative risk analysis. Annals of Operations Research

[8] Sulaman, Sardar Muhammad, Krzysztof Wnuk, and Martin H□2014. Perspective Based Risk Analysis-A Controlled Experiment. International Conference on Evaluation and Assessment in Software Engineering (EASE 2014).

[9] Ferrucci, Filomena, 2013. "Not going to take this anymore: multi-objective overtime planning for software engineering projects." Proceedings of the 2013 International Conference on Software Engineering. IEEE Press.

[10] Hu, Yong, 2013. Software project risk analysis using Bayesian networks with causality constraints. Decision Support Systems

[11] Hu, Yong, 2013. An integrative framework for intelligent software project risk planning. Decision Support Systems

[12] Heidrich, Jens. 2013. Software E ort Estimation and Risk Management. Product-Focused Software Process Improvement. Springer Berlin Heidelberg.

[13] Wautelet, Yves, Manuel Kolp, and Stephan Poelmans. 2013. Requirements-driven iterative project planning. Software and Data Technologies. Springer Berlin Heidelberg.

[14] Hsu, Wen-Ko, 2012. Risk and uncertainty analysis in the planning stages of a risk decision-making process. Natural hazards.

[15] He, Yong-xiu, 2011. Risk assessment of urban network planning in china based on the matter-element model and extension analysis. International Journal of Systems

[16] Asnar, Yudistira, Paolo Giorgini, and John Mylopoulos. 2011. Goal-driven risk assessment in requirements engineering. Requirements Engineering.

[17] Eric, S. K., Paolo Giorgini, and Neil Maiden, eds. 2011. Social modeling for requirements engineering. Mit Press.

[18] Zwikael, Ofer, and Mark Ahn. 2011. The effectiveness of risk management: an analysis of project risk planning across industries and countries. Risk analysis

[19] Jørgensen, Magne, and Kjetil Moløkken-Østvold. 2006. How large are software cost overruns? A review of the 1994 CHAOS report." Information and Software Technology

[20] Shruti Patil and Roshani Ade. 2014. Software Requirement Engineering Risk Prediction Model. International Journal of Computer Applications.

[21] Shruti Patil and Roshani Ade. 2014. Secured Cloud Support For Global Software Requirement Risk Management. International Journal of Software and Application.

[22] Shruti Patil and Roshani Ade. 2014. Cloud Data Security for Goal Driven Global Software Engineering Projects. International Conference on Information and Communication Technologies (ICICT 2014) Procedia Computer Science, Elsevier.

[23] Shruti Patil and Roshani Ade. 2015. A Software Project Risk Analysis Tool Using Software Development Goal Modeling Approach. Systems Design and Intelligent Applications Proceedings of Second International Conference INDIA 2015, Information Systems Design and Intelligent Applications , Volume 2, Series: Advances in Intelligent Systems and Computing , volume 340 ; Springer Verlag.