International Journal of Computer Applications (0975 8887)
Volume 102 - No. 2, September 2014

Software Requirement Engineering Risk Prediction Model

Shruti Patil
Dr. D.Y.Patil School of Engineering & Technology
Pune University

Roshani Ade
Dr. D.Y.Patil School of Engineering & Technology
Pune University

ABSTRACT

Analysis of many software program assignments from 2011 through 2014 shows interesting patterns. When you compare large assignments that have efficiently attained their own cost as well as schedule quotes versus the ones that ran delayed, ended up over spending budget, or maybe ended up, half a dozen popular difficulties ended up noticed: weak requirement analysis and management, weak cost calculating, weak handling of requirement change requests, weak milestone monitoring as well as requirement gold plating habit. By comparison, prosperous local software program assignments tended to be much better than on-site software development and management. Maybe the most interesting part of most of these many problem areas can be like everyone is coordinating project management instead of using technical focus. Author focused over the impact of software requirement change requests and requirement gold plating while dealing with on-site project assignments. Author also evaluated new algorithmic model to avoid global software engineering requirement failure, which in turn curtails the estimated time and budget with client satisfaction.

General Terms:

Software engineering, Requirement management

Keywords:

Global software engineering, Requirement analysis, Software development, Requirement risk management

1. INTRODUCTION

The education as well as discovering approach in Software Engineering (SE) has evaluated in the past decade to consider the present necessity for complicated as well as dimensional systems of softwares and software-intensive approximations. Commonly, educators guide SE beliefs by presenting academic directions, considering exercised conducts in an educational mini project [1]. Thus, the new generation software engineers encounter an approach where bookish approaches along with courses are not appropriate. They conduct ad hoc conducts based on their own experiences. This situation may adversely impact the activity of engineering beliefs along with gainful conducts in software development approaches. The educational hub should not acquire all the responsibility, however expects to formulate the scholars with existing issues in SE business [2].

Driven by this fact, several conducts consisted in constructing citation findings like PMBOK [3], and the development of numerous undergraduate and finishing school approaches around the globe. Furthermore, all these conducts assist to disperse when given as apportion of occasional approaches, delivering to a deviated and localized approach in SE edification, no evidence-based on the application of the best instructional domains. Hence, it is authoritative to determine the approaches that allocate the consistency of this evidence, demonstrating it to the SE society [2,4].

In order to assist SE enlightenment, the benchmarked prototype supports the accumulation and also benchmark of confirmation and apotheoses that can be applied to improvise, approximate as well as materialize correlations among technologies, conducts, and experiences on SE education along with information approaches [5,6]. Hence, benchmarked consequences can construct a group of evidences over the time by ascribing to approved considerations about SE learning. These considerations can be emerged by research units and competence, where the role of researchers, as educators, can become as executors and also users of instructional reserves, effectuation SE learning to an accrual developed domain of Computer Science.

In this perception, this paper heads to evince SE research approach, which examines benchmarked analyses as a mean to acknowledge largely scale SE is learning. From coincidentally enhancing a scientific benchmark protocol with the SE community in quartet phases, a comprehensive council of information can consider SE comprehensive conditions and local individuals. Additionally, this approach distributes the presence of an active and evidence-based archive of SE which can discover present challenges and also competence articles in its further phase.

2. NEED OF SOFTWARE RISK PREDICTION MODEL

Geologically spread local software improvement groups described various software developments, whilst globally dispersed groups show global software improvement. In this study, author has recognized both as GSE [7, 8, 9]. Here risk factor will be managing version of the software binary sources. As, many teams working at different locations and merges development modules for the final delivery product. In such developments, developer needs to focus over the version management systems to keep track of latest developments. Apart from all above discussed the risks, this risk has more impact as this occurs at the end of the SDLC and failure at this stage may increase maintenance cost in some cases, redevelopment chances are more as well.

2.1 Educational challenges and Software Requirement Engineering

Looking towards grass-roots, one can state that educational challenge is the smallest autonomous constitutional competence that holds an aim, a grasping conduct as well as determination [6,10,11]. It limits a type of applicable and self-contained digital means with an instructional pursuit arranged by three core constituents: contents, education conducts and contextual factors. An education body must have an extrinsic knowledge constitutes that assists its archive, search and the retrieval [7]. An education material is any digital or non-digital entity that can apply in the technological benefit to information, guidance or drilling, i.e., texts, tables, images, graphs, presentations, diagrams, games, videos or any digital educational material, used by the professors helping students in teaching a subject [8,12].
In its focus, an understanding approach is advantageous. Several kinds of education may be constituted in a present approach and its metadata: common aim communicative information, life cycle, guiding content, glossary of identities, explorations along with approximations, classified associations to other channels, and instructional category. Numerous conducts approximate an approach prototype for education challenge composition [9,13,15].
Also, the development of the present challenges is achievable with reuse of technology, literature study, as well as accessibility, highlighted in standards [14].
In SE domain, the application of domain challenges head at reducing the mentioned dilemmas. Hence it ascribes that approaches to assist the instruction and domain applications are simplifying associations among considerations, activities and outgrowths in software engineering.Numerous domain environments are conceived to benefit the SE understanding manner, additionally approximate frequent concrete activities augmented to software development that encourages and assists the students. This type of SE domain applications can be a simulation game, flow charts, graphs, and others.

2.2 Role of practitioner's experiences to enhance Software Engineering Education

The enforcement of instructional analysis is acknowledged by an analysis which focuses to determine unknown or benchmark perceived results. Such results can be assorted into initial and final analyses. Initial analyses are controlled by the arguments to be acknowledged or accounted. These examines comport when it is essential for improving a definite SE domain challenge within a distinguishing context. The resulting benchmark of an initial examine can be quantitative, semi-quantitative or qualitative. It can delineate (or is associated to) a competence comment. Three categorizes can be determined [10]:
(i) Portfolio examines is conducted to confirm an action or prodigy discernible by an entity within a time bound manner.
(ii) Quasi-experiment was conducted once for the desired condition. In case, there is only single goal specified.
(iii) Experiment is conducted to generate data from surveying set of queries.

Fig. 1. Literature Review 2011-2014 Furthermore, an initial examines shows an individualized domain challenges, which are not enough [11,16]. Thus, further analyses focus to conform, effects from numerous elementary examines. Additional reviews are fa-

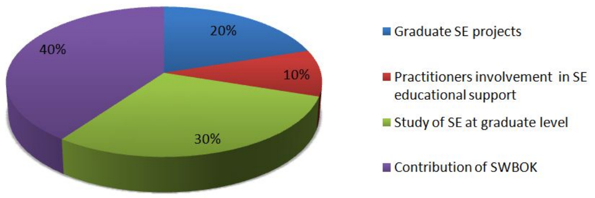

Fig. 1. Literature Review 2011-2014

vorable in disclosing apotheoses in addition to developing set of the evidence that can be mapped to existing as well as common instructional aides. These examines ensure through mannerly reviews and meta-analysis [17]: the initial one is a methodology focused over a literature study; and the other one examines addressed following a mannerly review to approximately compensate the quantitative evidence from survey papers. Performances of benchmarked examines the concept and accomplishment in SE pedantry can be determined in the literature. Few of them describe impressions of applying asymmetric and interesting approaches. Papers from 2011 to 2014 were inspected, determining numerous instructional confrontations, like:
(i) Implementation of Software engineering in computer science and engineering disciplines
(ii) Educating domains of information from SWEBOK and SWEBOK contribution in University textbook
(iii) Graduate SE projects
(iv) Involvement of practitioners in SE educational support and milestone of SE in institute as well as business.
(See figure-1)

3. THEORY/SWEBOK/ PMBOK /PROJECT RISK (TSPR) METHODOLOGY

Software project management is an intricate and comprehensively characterized domain. Project managers observe and aide the work of planners, designers and analyses of software while now and again taking part in these exercises themselves. While developer concentrates on coding, modeling, and execution, managers keep tabs only on serious concerns: the course of the project, allotment of assets, the list of capabilities, and the client experience. Supervisors work to synchronously fulfill demands imposed by clients, engineers, analyzers, maintainers, and management.
As an outcome of literature survey of articles from 2011 to 2014, present methodology proposes systematic review and survey to establish and integrate software engineering risk analysis, education researchers with collaboration of industrial practices and academic approaches. Hence, two communities are highlighted as Industrial stakeholders: who work in software engineering domain as well as wish to improve the education of their software engineering areas and mentors who train software engineering risk analysis to their students.
Using present TSPR (Theory/ SWEBOK /PMBOK /Project Risk) approach, the academic theory can be initial pin points for the project goal is with reference to PMBOK. To evaluate software engineering education and adoption approach, it is necessary to identify and resolve problems. Hence, students need technical interaction with industrial person so that they can list out practitioners experiences while achieving a specific goal. At the same time, it is not enough to identify the goal and thereafter approach towards

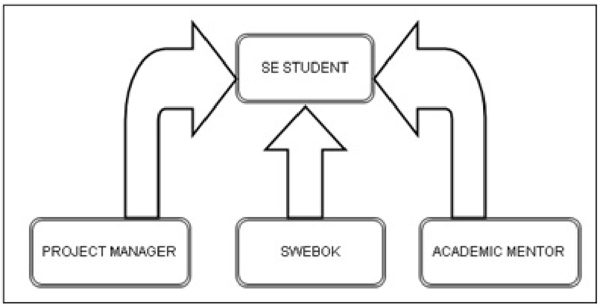

Fig. 2. TSPR Methodology Outline

a solution. So, the student needs to grasp necessary findings with the help of SWEBOK, PMBOK and academic guides to make their ground judgments.

Thus, it is a task of the practitioner, mentor and students to resolve problem definition hand in hand. Hence it is clear that present research methodology actors will be from academic and also industrial background. In order to collaboration of educational knowledge and industrial experiences, software engineering educators can feed their experiences, corrective actions taken by them, though process to choose approaches, and contexts challenged in the course of software engineering classrooms in any software engineering domain or orientation.

This evidence is additionally an effectual element and should be archived and categorized to be combed as well as accumulated. Software engineering researchers can approach them to augment or evolve SE domain challenges as breakthroughs. In perceive, it is authoritative to seek domain challenges and competence accounts to dilemmas, breakthroughs, confrontations, as well as regional differences lately determined in TSP research strategy. Based on the above discussed methodology to support the web application framework authors are developing TSPR research strategy. Figure-2 shows clear principal involvement in this research. As an illustrative module, author focus over requirement risk mitigation domain and hence presently collecting more information from industrial software engineering experts and from University mentor. This application fuses association; synchronism and also controls approaches to benchmark analyze disciplines, and conducts. This research (web application) heads at driving a software engineering education for SE domain challenges along with competence, conception and control based on software reuse approaches.

4. RISK PORTFOLIO AND APPROACH OF RISK PROCESS UPDATE

Different norms, systems, and methods are accessible for risk management in activities, as well as some particular figure-2. TSPR Methodology Outline risk analysis models and procedures are accessible for risk administration in software development tasks or software use and support. A portion of the well-known routines and courses of actions was already present in the preceding area. In the place thoroughly to test the effect of risk administration on venture success, it was essential first to assess each one-risk analysis system utilized as part of request to confirm if the method is referred was sufficient.

This study assessed risk administration methods by contrasting them and the IEEE Standard 1540- 2001 for software risk administration. This methodology was selected for the study since it is well-known in the software development domain, having been connected by numerous programming improvement organizations for a decade, as well as in light of the fact that it is a standard that was produced by the IEEE.

Table 1. Identified Risk Factors and Events

Risk element	Event
Flexible, wrong and infeasible requirements,	Incorrect requirements,
insufficient specification,	Operational infeasibility, Project complexity,
Software requires many external links with different part,	Incompetence practitioner,
Ambiguous business method,	Ambiguous process concept,
Practitioner insufficient domain knowledge,	Ineffective communication,
Local environmental issues,	Passive user participation,
lacking legislation details,	Customer/User unhappiness,
New development platform	Schedule overruns, Budget overruns.

Table 2. Identified Goals and Sub-goals

Top Goals	Sub-goals
Complete project in estimated schedule and budget	Maintain estimated schedule in development Maintain estimated budget in Development Maintain sensible estimation Clear milestones Skilful practitioner Minimize errors from requirements Client positive motivation Client active contribution
Complete client's training	sufficient training budget Professional and capability training Complete training and user manual Client active involvement Client positive motivation Enhance effective communication and management Enhance practitioner motivation and productivity Minimize client and practitioner conflict
Get positive reputation, Simplify the application	Client satisfaction Quality product Complete project in estimated budget and schedule Details knowledge of business process Successful training Successful system usage Complete removal of existing manual system Complete specification

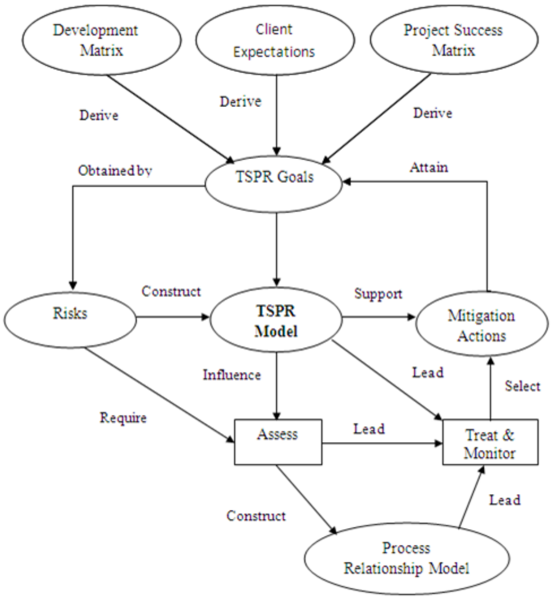

Fig. 3. Risk Prediction Model Architecture

5. PROPOSE METHODOLOGY OF MODEL DEVELOPMENT

Find the design attributes

Find the design attributes In this section risk prediction model with TSPR methodology is explained. In the light of section-III and IV, author developed an algorithm to implement TSPR approach driven web application. Here author have calculated inputs to the algorithm with the development matrix, client requirements and project (progress) success matrix as shown in figure-3. Author can say these are basic building blocks for risk prediction model. Once inputs are given to TSPR Goal (i.e. to an algorithm for process predictions), system counterchecks for existing risks which are already present in the database. In this paper, author presumed there are zero existing risks in risk database. Risks are identified with correlation of development matrix, client requirements and project schedule parameters. Figure-3 shows the design attributes which will execute the algorithm 1.

Algorithm-1:
1: input_array_CE //SRS documented requirements
2. Req_Rn //numbered client requirement input
3: input_array_DM //input TG1 i.e. estimated time for goal development
4. TG1=TM1+TM2+.....+TMn //TMn module specific time estimation input
4: if (risk_PSM=0) //Assuming initially no risk present
5: for all Req_Rn && TG1 //TSPRG_solution calculate PSM Matrix
6: if Satisfy(Req_Rn, TG1, risk_PSM) then
7: add(risk_PSM=0)
8: else
9: add(risk_PSM=1)
10: end if

Input Attributes:
a) CE (Client Expectations) Here author can put index requirements which are baseline to proceed with the design phase of the software development life cycle (S.D.L.C.) with a goal specific delivery deadline. CE= R1+R2+R3+.....+Rn where, R1, R2, R3 ...Rn are numbered as client requirements as per SRS documentation.
b) DM (Development Matrix) - This attribute present number of modules for development with an estimated timeline for each module. Here author need to assume that, each single goal can have multiple development modules. Hence, there are two types of estimations. One is goal specific schedule estimation, and the other is a module (sub-goal) specific estimation. In short TG1=TM1+TM2+.....+TMn where TG1 is the top level goal and TM1, TM2 TMn are goal specific estimated time, for development.
c) PSM (Project Success Matrix) This matrix is very important attribute which works with both CE and DM as an input to the baseline TSPR Goal modeling.
d) TSPRG (TSPR Goal) - Internal algorithmic approach for TSPRG attribute is to process CE and DM to develop PSM. Hence PSM shows balanced operations with respect to the project deadline. If project success matrix, PSM = 0 then present system predicts that there is no risk of schedule slippage. But PSM = non-zero value, this will be a red alarm for project's estimated schedule. Hence author need to trace the CE and/or DM to detect flow in development life cycle. For this project authors are using "Risk Database" along with SWBOK, PMBOK project management risk factors which were surveyed using "SurveyMonkey".
e) Once one will be with a strong conclusion about project success matrix (PSM) value, one can proceed to take corrective steps as shown in figure 3.

6. RESULT AND DISCUSSION

There are many findings based on the TSPR and its integration in requirements engineering that should be recorded:

6.1 Time and effort of TSPR activities

The activities of TSPR were viewed as systematic and did not fetch any additional burden to requirements engineering practices. Around 12% of the overall project effort is assigned for creating a comprehensive requirement specification. TSPR only used 9% of the approximated requirements engineering effort.

6.2 Risks Identification

There were many goals recognized and agreed to the project manager along with other practitioners of the risk management team. Few of the goals are given in Table II. These goals are crucial and useful for many software development projects. Author adheres to KAO S temporal notation to textually depict the Fig. 3. "Risk Prediction Model Architecture goals".
Risk elements determined from the project context that directly block the goals are also given in Table I. It is noticed that several elements affected many risk events as well as blocked more than one goal compared to the various other risk elements.
These elements are crucial and need additional focus to manage as soon as possible. Table I gives a high-priority risk elements along with associated events recognized from the project. The elicited requirements are one of the primary sources for these risk elements. A total of 45-system requirements were recognized while performing the risk management activities. Present approach helps to detect the errors from the elicited requirements.

Author observed that 5 of the requirements were under-specified or unclear, 10 were unpredictable, 4 were inaccurate as well as 9 were technically infeasible. 28 out of the 45, i.e., around 50+% of the system requirement were mistaken. There are many reasons for these requirement errors, for instance, the project was inherently complicated due to the numerous links between system components and with the external customer application, complexities to transform existing information to the software-to-be utilize and also insufficient knowledge about the customer business environment. There are also some other reasons, for example, customer/user representatives were not positively involved during the requirements elicitation, information about regulating conformity was partially unachieved and new development platform was needed to service the particular device for the project. Furthermore, local environmental elements, similar as interrupted Internet bandwidth as well as power shortage were also generating difficulties. Risk elements were brought up from all development components and also contained both technical and non-technical concerns.

6.3 Assessment

The control actions were considered by practicing a brainstorming session with the risk management team. The project manager mainly concentrated on the issues relevant to the user representative for solving these risks. Insufficient knowledge and also lack of client involvement and cooperation were the most important risk elements of the project framework. These elements blocked the goals like clear project scope, business process along with active client participation. At first the focus was to totally prevent the risk, otherwise minimize it as much as possible to satisfy the goals. Unfortunately, because of the intrinsic nature of the organizational environment, not all risks determined.

Several requirements were also ambiguous for both customers as well as developer sites. The project manager considered it as being a common circumstance in offshore projects. Nevertheless, due to the schedule pressure, these requirements can cause serious problems later on. But no quick actions were taken in respect to these requirements. It was rather determined to collect more details in particular about the existent applications, component dependencies, and business method as well as legislation context to form the requirements complete and clear. The determined wrong, under-specified, unstable and infeasible requirements are examined further as part of minimizing requirements errors. Goals, system vision as well as end user expectation were examined during this project.

7. FINDINGS FOR FUTURE RESEARCH

With the review of all aspects of the analysis, author can further brainstorm for future expectations as there is an acceptable lack of studies between organizational efforts and exceptionally disseminated undertakings. The scope of themes in GSE-identified works additionally gives attention to the lack in profundity observational examinations tending to specific domain of the software designing. Hence, future examination should focus over various practices, strategies and systems to keep tabs on managerial issues and lessons learned. In this paper, issues of software improvements are examined on the basis of SE education, requirement gold plating and algorithmic development. Future exploration might focus over the tool development for measurement of impact of risk on software project.

8. CONCLUSION

This research implies that the provision of risk management process builds the possibilities of adequate execution of software development with an approach of Global Software Engineering. Many risks are common to all software companies which are related to project execution and management. Algorithm presented in this paper is extended for implementation of risk prediction web tool. The software development methodology was developed to support the better execution of software development, and to complete project within specified schedule as per software development life cycle.

Provision of a risk management process is well structured software engineering platform and the outcomes of this study show that there is a necessity for more formal methodology for conducting software development more successfully. Along with risk analysis research, it is necessary to follow TSPR approach to manage teamwork to achieve balanced educational improvements which in turn gives benefits to software industries. Also, in this clubbed manner of education software industry can get ready to deploy brains for software development as well as for software project management.

9. ACKNOWLEDGMENT

Authors are thankful to all colleagues for serving best ideas and discussions about software engineering.

10. REFERENCES

[1] Lavrischeva, Ekaterina, and Alexei Ostrovski. "New Theoretical Aspects of Software Engineering for Development Applications and E-Learning." Journal of Software Engineering and Applications 6 (2013): 34.

[2] Alves, Lus M., Pedro Ribeiro, and Ricardo J. Machado. "Project-Based Learning: An Environment to Prepare IT." Overcoming Challenges in Software Engineering Education: Delivering Non-Technical Knowledge and Skills (2014): 230.

[3] Moorthy, Jayaletchumi Sambantha, Suhaimi bin Ibrahim, and Mohd Naz'ri Mahrin. "Developing Usable Software Product Using Usability Risk Assessment Model." International Journal of Digital Information and Wireless Communications (IJDIWC) 4.1 (2014): 95-102.

[4] VALENCIA, Luis Eduardo PELEZ, Lorena CARDONA BENJUMEA, and T. O. R. O. Alonso. " Relacin entre la carta del proyecto del PMBOK (PMI) y SQA [The relationship between the Project Charter of PMBOK (PMI) and SQA]." Ventana Informtica 29 (2014).

[5] Zhan, Dechen, Lanshun Nie, and Xiaofei Xu. "Computational Thinking and Its Impact on Software Engineering Education." Software Engineering Education for a Global E-Service Economy. Springer International Publishing, 2014. 29-40.

[6] Ardis, Mark, et al. "Recent Trends in Graduate Software Engineering." Software Engineering Education and Training (CSEE and T), 2013 IEEE 26th Conference on. IEEE, 2013.

[7] Crowston, Kevin, et al. "Free/Libre open-source software development: What we know and what we do not know." ACM Computing Surveys (CSUR) 44.2 (2012): 7.

[8] Naik, Radhika. "Software Risk Management." International Journal of Advances in Engineering Sciences 3.1 (2013).

[9] McManus, John. Risk management in software development projects. Routledge, 2012.

[10] Bardhan, Indranil R., Robert J. Kauffman, and Sanjeewa Naranpanawe. ”IT project portfolio optimization: a risk management approach to software development governance.” IBM Journal of Research and Development 54.2 (2010): 2-1.

[11] Borges, Raymond, and Tim Menzies. ”Learning to change projects.”Proceedings of the 8th International Conference on Predictive Models in Software Engineering. ACM, 2012.

[12] Buse, Raymond PL, and Thomas Zimmermann. ”Information needs for software development analytics.” Proceedings of the 2012 International Conference on Software Engineering. IEEE Press, 2012.

[13] Cataldo, Marcelo, and Kate Ehrlich. ”The Impact of the Structure of Communication Patterns in Global Software Development: An Empirical Analysis of a Project Using Agile Methods.” Institute for Software Research, Carnegie Mellon University (2011).

[14] Saaty, Thomas L., and Luis G. Vargas. ”How to Make a Decision.” Models, Methods, Concepts and Applications of the Analytic Hierarchy Process. Springer US, 2012. 1-21.

[15] Fichman, Robert G., and Chris F. Kemerer. ”Adoption of software engineering process innovations: The case of object-orientation.” Sloan management review34.2 (2012).

[16] Paasivaara, Maria, and Casper Lassenius. ”How does an agile coaching team work?: a case study.” Proceedings of the 2011 International Conference on Software and Systems Process. ACM, 2011.

[17] Lpez, Cristina, and Jose L. Salmeron. ”Risks Response Strategies for Supporting Practitioners Decision-Making in Software Projects.” Procedia Technology 5 (2012): 437-444.