I.J. Information Technology and Computer Science, **2015, 06, 38-44**
Published Online May 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.06.05

Proposal of New PRORISK Model for GSD Projects

M. Rizwan Jameel Qureshi, Aysha Albarqi
Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: anriz@hotmail.com, albarqiaisha@yahoo.com

Abstract— The level of complexity and risks associated with software are increasing exponentially because of competing environment especially in geographically distributed projects. Global software development (GSD) face challenges like distance, communication and coordination challenges. The coordination and communication challenges are the main causes of failure in GSD. Project Oriented Risk Management (PRORISK) is one of the models to address the importance of risk management and project management processes in standard software projects. However, existing model is not proposed to handle GSD associated risks. This warrants the proposal of new PRORISK model to manage the risks of GSD. Survey is used as a research design to validate the proposed solution. We anticipate that the proposed solution will help the software companies to cater the risks associated with GSD.

***Index Terms*—** Global Software Development, Distributed Projects, Risk Management, Risk Mitigation**.**

I. INTRODUCTION

Global Software Development (GSD) is the most recent domain in software engineering [1]. GSD is a software development that is distributed across two or more sites. The sites may be separated by national or continental borders [2] where developing software project across distances is familiar today. The various advantages provided by GSD for organizations that develop software are reduction in marketing time, improvement in quality, reducing cost and increasing productivity. However, GSD faces various challenges that are not exist in the traditional collocated software development project these challenges could be in control, communication and coordination according to socio-cultural, geographical and temporal distance [1]. The classification of these challenges is mentioned in [2] and these challenges must overcome to take advantages of GSD.

Agile software development has gained too much interest in last several years due to its flexible approach towards managing requirements, fostering of close collaboration between developers and customers, and frequent and early and reduced risks about failure of software. Agile methods are proposed assuming co-located development teams while GSD assumes distributed teams. Practitioners are adapting agile practices to GSD to take advantages of both methods. Scrum method is gaining more attention and it is extensively practiced in GSD projects as compared to the other agile methods [1].

Geographically distributed teams work with shared purpose across space, time, and organizational boundaries and use technology to communicate and collaborate [1]. Software development is complex and risky by its nature. Complexity and risks are further increased in GSD projects. Risks are the possibility of loss where software development projects are full of risks, survey paper show that 15 to 35 percentages of software development projects have been canceled halfway because of risks whereas the rest of the project cannot achieve the desired objectives or not extended beyond the budget. It is suggested that 90% risks can be avoided if the risk management process is applied on projects [3]. Some risks are called dangerous or deadly risks and if deadly risks penetrate into project may affect or fail a project.

Software project risk is process of software project development due to cost, quality, and environmental uncertainties impact. All of these potential risks need to be managed to guarantee the success of software project development.

A huge number of statistics demonstrate the effectiveness of implementing risk management in development process of software project which this management process leads to reduce losses as possible. The process of management risks in software accomplished by team of project management through the identification, quantification, and control of risks, with using a several management tools, techniques, and methods [3].

Global software development (GSD) projects are accompanied with both opportunities and challenges. GSD faces numerous management challenges like coordination and communication and it is hard to implement risk management. There is pressing need of an efficient risk management model to identify, plan and control risks of GSD to solve the software industry problem.

Many software development projects miss their goals of delivering acceptable software products within agreed constraints of time, budget and quality due to lack of effective software risk management process. There are many models to manage risks such as Bany Boehm theory, MSF Risk Management Model, CMMI (Software Capability Maturity Model Integration), IEEE risk management standards, SEI’s (Software Engineering Institution) continuous risk management (CRM) model, Softrisk risk management model and PRORISK Project Oriented Risk Management model [3]. PRORISK is one of the risk management models for collocated teams. PRORISK model involves three main elements: project management, risk management, and risk database. The project management process is the core process of the project while the risk management process focuses on handling the risks and risk database act as bridge between the project management and risk management to connecting them and facilitate exchanging information between them. This paper will contribute in facilitate the risk management process by proposing new PRORISK model to manage the risks of GSD.

This paper is organized as follows. Section 2 describes related work to formulate the problem statement. Section 3 defines the problem and proposes new PRORISK model. Validation of the proposed solution is provided in section 4. The 5th and last section concludes the paper.

II. Related Work

Krishna et al. [1] described an approach to support risk management by identifying vulnerability and criticality risk factors at organizational level prior to their propagation and reflection at a process level. Suebkuna et al. [2] propose an improved risk management model by applying on a software project with collocated team. The proposed model needs to test on a distributed project to check its effectiveness in GSD. Tianyin et al. [3] reviewed research results in the area of risk management. Tianyin et al. [3] conclude that it is vital and urgent for enterprise companies to establish risk management system. Chen et al. [4] analyzed the process of enterprise risk management based on the theory and methods of spiral model. Chen et al. [4] conclude that the Spiral model improves risk management and brings into economic benefits due to ‘Risk Analysis’ phase [4].

Zheng et al. [5] discussed how to quantify the risk using Value at Risk (VaR) model. Kriouile et al. [6] aimed to contribute to the development of information systems (ISs). Kriouile et al. proposed a maturity model to manage risks of ISs. A case study is conducted to validate the proposed model. Sadat et al. [7] proposed a model to manage risks. The model is validated using a case study in an industrial context. Leung et al. [8] proposed a stochastic simulation model to manage risks. Leung et al. [8] developed a simulator to test the proposed model. Lin et al. [9] analyzed the schedule risk and established a GERT network model. Marques et al. [10] proposed formal control model to reduce the human dependency. A case study is conducted to validate the proposed model [10].

Gannon [11] explores the fundamentals of SCRUM as well as how this agile development methodology has been implemented on a project at the Johns Hopkins University Applied Physics Laboratory. The team members faced some difficulty in time management, co-located with the rest of the team and each team member's amenability to change [11]. Guang [12] has developed a vehicle management system using scrum as software process method with the template visual studio 2010 in order to improve the productivity and efficiency of SCRUM.

Łukasiewicz and Miler [13] proposed a C–S model to improve an actual software development process that defined a diagnostic questionnaire, a practice selection algorithm, an application process and built a software tool, these proposals implemented on two case studies. 3.5% of the suggested practices were rejected by the companies. Some 24.5% of suggestions were evaluated as not applicable for organizational or economic reasons, which leaves room for further improvement of proposed questionnaire. It is discussed that how to implement collaborative agile scrum software for complex multi-vendor competing environments [14]. A framework is proposed for collaborative agile software development. The roles of project manager, product owner and scrum master are specified. The specialty of this framework is that scrum master can work with product owner and expected to support organizational learning and deliver benefits at strategic level. Four different flexible agile methods are identified to transfer knowledge across projects:

- mentoring and coaching;
- staffing project teams with members of other projects;
- participation in multi-project reviews;
- anticipation in multi-project retrospectives.

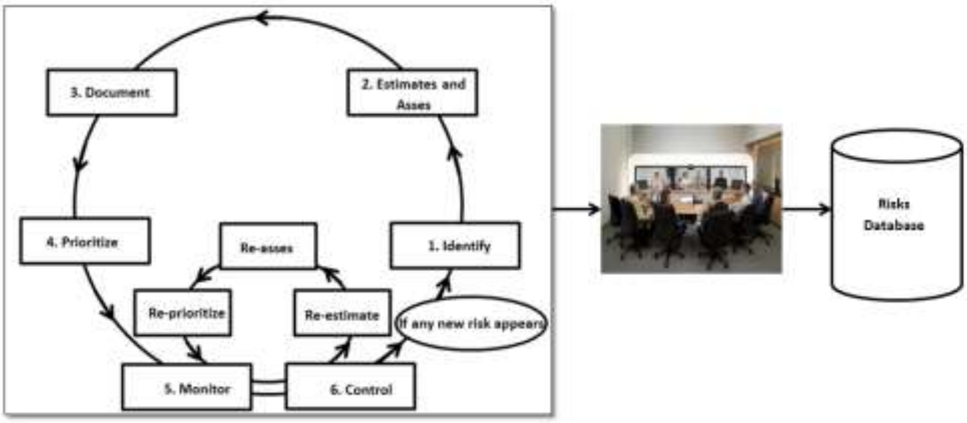

Fig. 1. The Proposed Risk Management Model

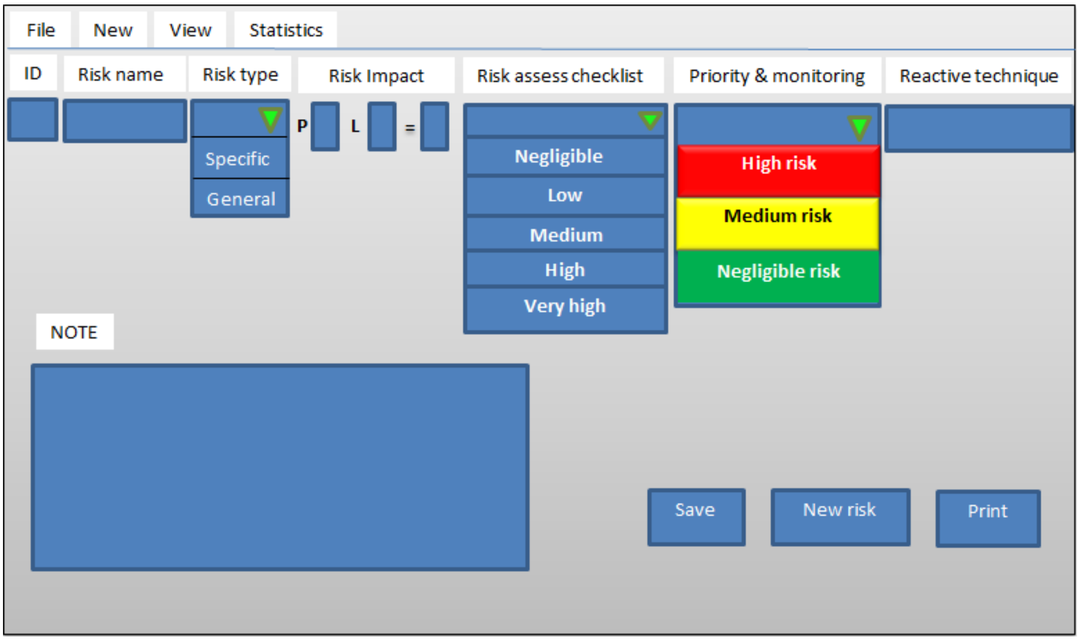

Fig. 2. The Proposed Risk Register Tool to Manage Risk details

These four methods can be used as a guide for development of strategies to promote learning across scrum projects [15]. Samina and Javed [16] framework is a customized approach to improve software development by integrating CMMI and scrum. Table 1 illustrates the summary of related work.

Table 1. Summary of Related Work

Title	Summary
Evaluating Effectiveness of Risk Identification and Management Using Organizational Models	Support risk management by identification of risk factors at organizational level
Towards A Complete Project Oriented Risk Management Model: A Refinement of PRORISK	Refined PRORISK model
Development of software project risk management model review	Reviewed research results of software project risk management
The Application of Spiral Model in Enterprise Risk Management	Analyzed the process of enterprise risk management based on spiral model
Risk management: VaR model for information disclosure	Discussed how to quantify the risk with VaR model.
Model ISR3Mfor assessing maturity of IS risk management process	Contributed to the development of information system governance and more specifically of IS risk management and proposed a maturity model for IS risk management
Rethinking the Mitigation Phase in Software Risk Management Process: A Case Study	Proposed model considered the typical phases of risk management by focusing on the unseen risks and opportunities accompanying with the risk mitigation decisions.
A Stochastic Simulation Model for Risk Management Process	Proposed a stochastic simulation model for risk management process
Schedule Risk Management at Early Stages of Large Construction Projects Based on the GERT Model	Analyzed the schedule risk at the early stages and developed a GERT network model
A Formal Control Model for Risks Management within Software Projects	Proposed formal control model to manage risks

Qureshi and Noha [17] worked on how Scrum methodology helps to mitigate the effects of temporal distance which includes increased coordination costs in GSD projects. A web application called (Distributed Scrum Web Application) provides various advantages for Scrum teams. The main advantage of this application is to facilitate communication among distributed team members [17]. Alotaibi and Qureshi [18] proposed an algorithm to adapt SCRUM practices to mitigate temporal distance-based GSD coordination challenges. Few strategies are discussed to mitigate the coordination challenges between distributed teams. The proposed algorithm is classified on the basis of the number of hours of work shared between sites. It will help Scrum teams in the identification and selection of appropriate strategies [18].

III. Problem Defintion and the Proposed Solution

Distributed development projects face many challenges and risks. PRORISK model is proposed to manage risks associated to collocated teams. PRORISK model needs to be extended due to ever increasing demands of GSD projects. PRORISK model works best for projects with collocated teams. Fig. 1 shows the proposed PRORISK model.

Risk management, risk database and project management are the three main components of proposed PRORISK model. Identify all potential risks related to GSD. Estimate the impact of each risk by multiplying risk probability and cost impact.

Risk Impact = P * L (1)

P = Probability of Risk
L = Loss/Cost Impact

A risk management plan is established. Risk management plan is implemented if the risk is occurring. A tool is proposed to identify and rank risks associated to GSD as shown in fig. 2. Risk management plan is developed by software development team to precede, contain and mitigate the effects of risk to a project. Each risk will be analyzed in order to understand its source, effects, probability of loss and how a team can prevent it. The management plan includes attributes such as a brief risk description, the data which identify the risk, probability of occurrence of the risk, degree of its impact, the person who manages, controls, and takes action in response to a risk, action against risk and risk status.

It is necessary not to lose agility of a project while integrating risk management with Scrum. We propose a ‘Risk Register’ tool to track and manage risks in Scrum projects. The proposed ‘Risk Register’ tool will contain risk assessments corresponding to a particular sprint to monitor that how many risks will be added and removed. Table 2 shows the attributes of the proposed ‘Risk Register’ tool. These attributes are:

1) risk description;
2) data identified causing a risk;
3) probability of a risk in percentage value;
4) impact on the project (It ranges from 1 to 5);
5) way to solve a risk;
6) allocate resources to handle a risk;
7) tatus of a risk;
8) exposure;
9) priority high means manage first.

Table 2. The main Attributes of Risk Register Tool

Risk description	Data identified	Probability	Impact	Solution/Action	Owner	Status	Exposure	Priority
Security	Access risks	40%	5	Termination Work Flow Plan	Project Manager/ Scrum Master		10	5
Schedule	Complexity, size and syntax/logical errors	25%	4	Use CBD and other extra resources	Scrum team		4	4
Environment	Natural event	5%	2	back up the project andimplement telecommuting	Scrum master		0.2	1
Incorrect or missing functions	Ambiguity in requirements	20%	3	Validate and verify requirements using prototype	Scrum team		1.8	3
Unstable requirements	New requirements	10%	3	Extra cost and time is required	Scrum owner		0.9	2

The proposed ‘Risk Register’ will be used in each sprint to add suspected risks. Scrum team will use the ‘Risk Register’ component to deal with the risk before the implementation phase of a sprint. Team and ‘Scrum Master’ will review the risks using the proposed ‘Risk Register’ during sprint retrospective meeting.

IV. Validation of the Proposed Solution

Survey is used as a research method to validate the proposed solution. The questionnaire is distributed among ten software development companies who are involved in GSD. Two questionnaires are designed to evaluate the proposed solution. First questionnaire is prepared to validate risk register component. Table 2 shows data of first questionnaire to evaluate the proposed ‘Risk Register’ tool by taking the average.

A. Security Risk

There three two common security risks reported during this survey. Firstly, the employees resigned or retired still have access to systems. Secondly, more than needed users have access rights. Thirdly, changes management process is not implemented. 40% probability was identified and impact of this risk is very high i.e., 5. Termination work flow plan is suggested to mitigate the risks. Project Manager/Scrum master is responsible to implement the plan as risk is monitored. Risk exposure is 10 out of 10 companies where survey is conducted and priority is very high i.e., 5.

B. Schedule Risk

Late delivery of software risk is due to complexity, size and syntax/logical errors. 25% probability is estimated with high impact. The component based development is suggested to develop a repository to reuse code as a plan. The exposure and priority of this risk is 4.

C. Environment Risk

Environment risk is raised due to fire, flood and bad weather. 5% probability is estimated with low impact. Implement telecommuting technology and back up the data periodically are the suggested solutions to manage the environment risk. The priority of this risk is very low with 0.2 exposures.

D. Incorrect or Missing Functions Risk

Ambiguous or unclear requirements are the cause of incorrect or missing functions risk with 20% probability, nominal impact, 1.8 are exposures and nominal priority. Use of heavy prototype is the suggested solution to validate and verify requirements.

E. Unstable Requirements Risk

Unstable requirements risk is raised due to frequent change in requirements by customer. 10% probability of this risk is reported by the software companies with nominal impact. Extra cost and time is required to manage the unstable requirements. 0.9 is the exposure of this risk with low priority.

The questionnaire two is divided into three goals.

Goal 1: Importance of project management in GSD project. This stage takes care of communication, collaboration, coordination of GSD teams to identify responsibilities and duties.

Goal 2: Importance of risk management. This stage monitors and controls risks.

Goal 3: Identifying the best techniques and methods to cater risks in a GSD project.

a. Cumulative Statistical Analysis of Goal 1: Determine the importance of project management in a GSD project.

It is displayed in Table 3 that 78% of the respondents are of the view that 'project management' has very high effect on the development of GSD projects. It is also reflected that 17% of the participants are recommending that 'project management' has high effect on the development of GSD projects. 4% of the respondents report the nominal effect of 'project management' on the development of GSD projects. 1% of the software professionals report low and very low effect of 'project management' on the development of GSD projects. The graph in fig. 3 appears to be supporting the validity of Goal 1.

Table 3. Cumulative Statistical Analysis of Goal 1

Q. No.	VeryLow	Low	Nominal	High	VeryHigh
1	2%	2%	4%	20%	72%
2	0%	2%	6%	15%	78%
4	2%	0%	2%	17%	80%
6	0%	0%	6%	17%	78%
20	0%	2%	2%	15%	81%
Total	4%	6%	20%	84%	389%
Avg.	1%	1%	4%	17%	78%

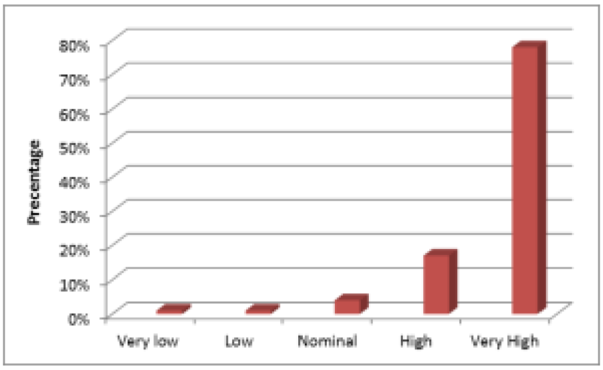

Fig. 3. Graph shows the cumulative results of Goal 1

b. Cumulative Statistical Analysis of Goal 2: Determine the importance of risk management in GSD projects.

Table 4 shows that 77% of the respondents are very highly favoring the importance of risk management in GSD projects. Among the software professionals, 17% of the participants are highly favoring the importance of risk management in GSD projects. 5% of the respondents report the nominal effect of this issue. 1% of the respondents are reporting the very low and low effect of risk management in GSD projects. Fig. 4 shows this graphically as follows.

Table 4. Cumulative Statistical Analysis of Goal 2

Q. No.	Very low	Low	Nominal	High	Very high
3	0%	0%	7%	20%	73%
5	0%	2%	5%	18%	75%
7	2%	0%	5%	16%	76%
8	0%	2%	2%	15%	82%
9	0%	0%	7%	15%	78%
10	2%	0%	2%	18%	78%
11	0%	2%	5%	16%	76%
Total	4%	6%	33%	118%	538%
Average	1%	1%	5%	17%	77%

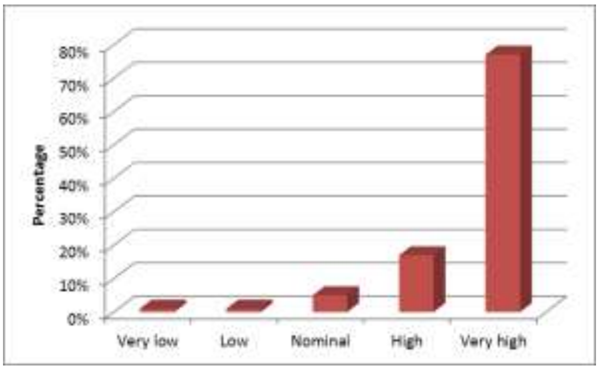

Fig. 4. Graph shows the cumulative results of Goal 2

C. Cumulative Statistical Analysis of Goal 3: Identifying the best techniques and methods to cater risks in a GSD project.

Table 5 shows that 79% of the respondents are highly favoring the 3rd goal. Among the software professionals, 16% of the participants are very highly favoring this goal.

4% of the respondents report the nominal effect of the respective goal. 1% of the software professionals report the very low and low effect of this goal. Fig. 5 shows the results of goal 3 graphically.

Table 5. Cumulative Statistical Analysis of Goal 3

Q. No.	Very Low	Low	Nominal	High	Very High
12	0%	0%	4%	18%	78%
13	2%	0%	4%	18%	76%
14	0%	2%	2%	16%	80%
15	0%	0%	7%	15%	78%
16	2%	0%	2%	16%	80%
17	0%	2%	4%	13%	82%
18	0%	0%	4%	18%	78%
19	2%	0%	4%	16%	78%
Total	6%	4%	31%	130%	630%
Avg.	1%	1%	4%	16%	79%

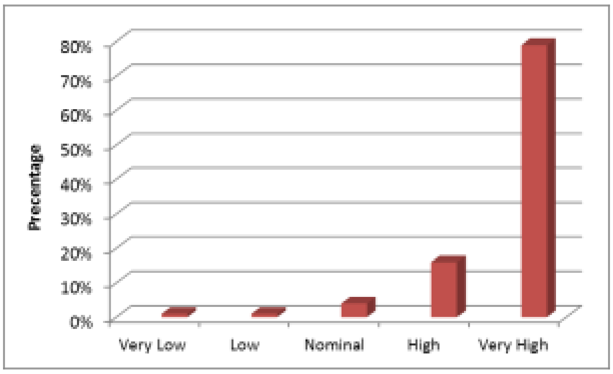

Fig. 5. Graph shows the cumulative results of Goal 3

D. Final Cumulative Evaluation of all Goals

The results of final cumulative statistical analysis of the three goals are shown in Table 6.

Table 6 shows that only 1% of the software engineers report very low effect of the proposed changes in the PRORISK model for GSD projects. 1% of the professionals respond the low effect of the three goals. 4% of the respondents are remained neutral. 17% of the respondents are highly favoring the goals 1 through 3. Among the software professionals, 78% are very highly agreed. As such 87% support is available. Fig. 6 shows the results of Table 6 graphically.

Table 6. Final Cumulative Evaluation of all Goals

Q. NO.	Very Low	Low	Nominal	High	Very High
1	2%	2%	4%	20%	72%
2	0%	2%	6%	15%	78%
3	0%	0%	7%	20%	73%
4	2%	0%	2%	17%	80%
5	0%	2%	5%	18%	75%
6	0%	0%	6%	17%	78%
7	2%	0%	5%	16%	76%
8	0%	2%	2%	15%	82%
9	0%	0%	7%	15%	78%
10	2%	0%	2%	18%	78%
11	0%	2%	5%	16%	76%
12	0%	0%	4%	18%	78%
13	2%	0%	4%	18%	76%
14	0%	2%	2%	16%	80%
15	0%	0%	7%	15%	78%
16	2%	0%	2%	16%	80%
17	0%	2%	4%	13%	82%
18	0%	0%	4%	18%	78%
19	2%	0%	4%	16%	78%
20	0%	2%	2%	15%	81%
Total	14%	16%	84%	332%	1557%
Avg.	1%	1%	4%	17%	78%

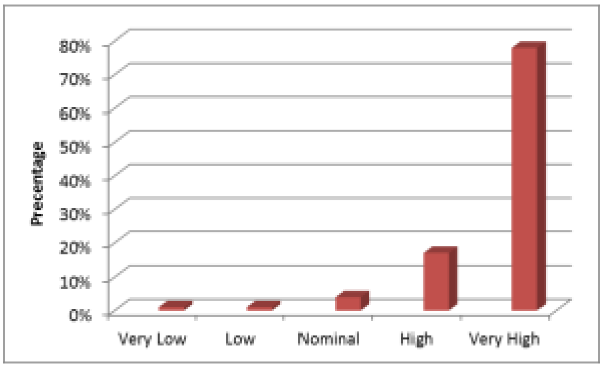

Fig. 6. Graph shows the cumulative results of all Goals

V. Conclusion

In this paper, the PRORISK model is proposed to implement in GSD projects. We also proposed a risk management tool. This is achieved by proposing a ‘Risk Register’ component to cater the possible risks regarding failure of a software project. The main attributes of the proposed ‘Risk Register’ component are defined during this research. Scrum team will use the proposed component during all sprint review meetings.

Survey is used as a research methodology to validate the proposed solution. The proposed solution is validated using two questionnaires. 1st questionnaire is validating the matrix of the ‘Risk Register’ tool. Second questionnaire is designed against three goals. The results are displayed using frequency tables and bar charts. A support of 87% is achieved against the proposed solution as shown in Table 6 and fig. 6. We anticipate that the proposed research will help to software companies who are involved in GSD to cater the coordination and communications risks.

REFERENCES

[1] Krishna, M. Bhuiyan, and S. Rana, "Evaluating Effectiveness of Risk Identification and Management Using Organizational Models," in Computer Research and Development (ICCRD), 3rd International Conference, 2011, pp. 278–282.

[2] P. L. Bannerman, E. Hossain, and R. Jeffery. "Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?" in*45th Hawaii International Conference on System Sciences*, 2012, pp. 5309-5318.

[3] P. Tianyin, "Development of software project risk management model review," in Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on: IEEE, pp. 2979-2982.

[4] Z. Chen, "The application of spiral model in enterprise risk management," in Electronics and Optoelectronics (ICEOE), 2011 International Conference. vol. 1: IEEE, pp. 376-379.

[5] Y. Zheng, "Risk management: VaR model for information disclosure," in Proceedings of the 2012 Second International Conference on Business Computing and Global Informatization: IEEE Computer Society, pp. 183-186.

[6] M. Elmaallam and A. Kriouile, "Model ISR3M for assessing maturity of IS risk management process: Case study," in Information Science and Technology (CIST), 2012 Colloquium in: IEEE, pp. 16-21.

[7] S. Khatavakhotan and S. H. Ow, "Rethinking the Mitigation Phase in Software Risk Management Process: A Case Study," in Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference on: IEEE, pp. 381-386.

[8] P. Zhou and H. K. N. Leung, "A Stochastic Simulation Model for Risk Management Process," in Proceedings of the 2012 19th Asia-Pacific Software Engineering Conference-Volume 01: IEEE Computer Society, pp. 737-742.

[9] X. Gao, Y. Shen, and J. Lin, "Schedule risk management at early stages of large construction projects based on the GERT model," in Proceedings of the Anthology Conference, IEEE, pp. 1-4.

[10] F. R. M. Cardoso, P. M. Tasinaffo, D. A. Montini, D. D. Fernandes, A. M. da Cunha, and L. A. V. Dias, "A Formal Control Model for Risks Management within Software Projects," in Information Technology: New Generations (ITNG), 2012 Ninth International Conference on: IEEE, pp. 452-457.

[11] Gannon, M., An Agile Implementation of SCRUM, Aerospace Conference, p. 1-7, 2013.

[12] Guang-yong, H., Study and practice of Import Scrum agile software development, p. 217-220, 2011.

[13] Łukasiewicz, K., Miler, J., Improving agility and discipline of software development with the Scrum and CMMI, IET Software, 2012. 6(5): p. 416-422.

[14] Ramanujam, R., Lee, I., Collaborative and Competitive Strategies for Agile Scrum Development, 7th International Conference on Networked Computing and Advanced Information Management (NCM), p. 123-127, 2011.

[15] Landaeta, E., Viscardi, S., Tolk, A., Strategic Management of Scrum Projects: An Organizational Learning Perspective, Int'l Technology Management Conference, p. 651-656, 2011.

[16] Samina Jan and Ali Javed, SCXTREME Framework: A Customized Approach of Process Improvements in Agile Blend with CMMI Practices in Pakistan, IJCNIS, 2013, 5(3): p. 69-78.

[17] M. Rizwan Jameel Qureshi and Noha Alsulami, “Mitigating Coordination Costs in Global Software Development Using Scrum,” International Journal of Information Engineering and Electronic Business (IJIEEB), online June 2014; 6(3), 2014, pp. 16-21.

[18] Asmaa Fowzi Alotaibi and M. Rizwan Jameel Qureshi, “Scrum and Temporal Distance-Based Global Software Development,” International Journal of Computer Network and Information Security (IJCNIS), online May 2014; 6(6), 2014, pp. 48-53.

Authors’ Profiles

Aysha Abdullah Albarqi: Post-graduate student in King Abdul-Aziz University. She received her bachelor of Computer Science in 2011.

Dr. M. Rizwan Jameel Qureshi received his Ph.D. Computer Sciences degree from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an associate professor in the Department of Information Technology, Faculty of Computing & Information Technology, King Abdulaziz University. He is the best researcher awardees of King Abdul-Aziz University Saudi Arabia in 2013 and Department of Computer Science, COMSATS Institute of Information Technology Pakistan in 2008.

How to cite this paper: M. Rizwan Jameel Qureshi, Aysha Albarqi,"Proposal of New PRORISK Model for GSD Projects", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.6, pp.38-44, 2015. DOI: 10.5815/ijitcs.2015.06.05