

Risk Management Adoption Framework for Software Projects: A Case Study for Kenyan Software Project Managers and Developers

Noela Jemutai Kipyegen¹, Waweru Mwangi² and Stephen Kimani³

¹Department of Computer Science, Egerton University,
P. O. BOX 536 Njoro, Kenya.

²Institute of Computer Science and Information Technology
Jomo Kenyatta University of Agriculture and Technology
P. O. BOX 62000- 00200 Nairobi, Kenya

³Institute of Computer Science and Information Technology
Jomo Kenyatta University of Agriculture and Technology
P. O. BOX 62000- 00200 Nairobi, Kenya

Abstract

Despite the existence and availability of proper software development risk management processes, Kenyan's software projects experience myriad of challenges. This study looks at software risk management practice and the root of problems experienced by Kenyan software project managers and developers. The survey helped to understand these causes and the kind of risk management adoption framework suitable to software projects and that solves existing problems in Kenya. Results from 62 software developers and 15 IT project managers from 30 software development firms showed that majority (87%) of IT project managers and developers do not use formal risk management techniques. There are no measures put in place by various development organizations for the implementation of formal risk management methods. This paper develops a framework that guides in the adoption of the existing formal risk management techniques in two areas; Institutions of learning and software development industry.

Keywords: *Formal Risk Management, Root, Risk Management Adoption Framework.*

1. Introduction

Formal software development risk management tools, techniques and standards exist but despite their existence and availability, risks still persist in software projects. Like majority of countries in Africa, Kenya's software development is a young and rapidly growing industry facing myriad of challenges. These challenges include delays, underperformance, cost overrun [1] and rework.

Risk management is the identification of hazards and possible problems, evaluation of their importance, drawing up of plans to monitor and deal with those problems [2]. These risks include incomplete software requirements, or volatile requirements, poor design, employees' turnover, lack of formal risk management approach, poor schedule

and cost estimates, complex projects, and lack of skills among others.

The need and motivation to undertake this study was conceived after observing the enormity of challenges faced by Kenyan software projects. The main objective was to understand the practice of risk management and the root of the problems experienced in Kenya's software industry. Another objective was to develop a framework based on this study finding.

There is a lack of framework in software development firms and institutions of learning on how to get started on the implementation of formal risk management techniques. Therefore, we propose a simple framework that enables various organizations adopt the practice of a disciplined, simple approach to risk management so as to reduce crisis management and the clean ups that may result [2] [3] [4] [5] [6] [7] [8]. It will also increase the chance of project success [9].

The objective of the framework is to provide a way of dealing with challenges faced by Kenyan software development firms. It combines Kenyan experience through a survey that was conducted to establish the state of risk management practices and the root of the problems experienced by software developers and IT project managers. This framework is supposed to provide guidelines on implementation of formal risk management techniques for software development both in the industry and institutions of learning. In addressing this issue, this paper addresses implementation of risk management practices at, 1) Institution of learning (computer science and software engineering institution); to equip upcoming developers with proper knowledge and 2) software development industry.

2. Theoretical Background

No project is free of risk [7]. Various reasons have been given for prevalence of risks. These include the fact that practitioners poorly understand the area of software engineering risk assessment even though it is an important issue, which can save millions of dollars in the projects, [10]. Another reason is the intangible nature and uniqueness of software, which means that traditional processes for managing industrial projects are not effective [11]. This makes it more difficult to plan, model and predict the progress of a software project [12]. Other reasons include; users' high expectations of the software, constant review of computer systems where new technologies replace the existing technologies which are obsolete and impractical for new developments. Market competition is also high, which requires organizations to keep constant pace with Information Technology (IT). These rapid changes and growing expectations have led to decreased stability and increased software development complexity bringing along new risks to the software projects [13].

The proposed risk management paradigms by Hall [2], Higuera et al [3], McManus [7], Boehm [14], IEEE [15], Carr et al [16], Williams et al [17] comprises almost similar components and processes, such as; risk identification, risk analysis, risk prioritization, risk management planning, risk resolution and risk monitoring. For example SEI's paradigm, Figure 1, is structured around a set of continuous tasks that guide the risk management process. The paradigm elaborates and illustrates the steps that are identified as continuous activities throughout life cycle of the project. This process can be tailored to the specific environment of a software project [3] [18] [19] [20] [21].



Fig. 1 SEI Paradigm. Source: William [17]

“Risk identification is the first step in the software risk management process. Risks are identified and added to the list of known risks. After identification, risks are then analyzed, which is the conversion of risk data into risk decision-making information. Analysis provides the basis

for the project manager to work on the right and most critical risks” [18].

Planning involves turning risk information into decisions and actions. Risk tracking consists of monitoring the status of risks and the actions taken to ameliorate them. Risk control corrects deviations from planned risk actions. Risk communication lies at the center of the model to emphasize both its pervasiveness and its criticality. Without effective communication, no risk management approach can be viable [18]. This includes communication within levels of project development and organization, the customer's organization, and most especially, across that threshold between the developer, the customer, and, in other cases, the user.

3. Methodology

The commencement point for this study involved understanding risk management concepts and acceptable practices found in the existing literature. The surveyed literature covered project risk management and lessons learned from other surveys. These literatures helped us achieve both breadth and depth in the area of software risk management, contributions, contexts, and general current state of risk in software development industry.

We strongly believe that to be able to address software project's issues effectively, it is important to understand how risk management is currently being practiced in Kenyan software projects. Therefore, this research is based on collected data which is then analyzed and organized to reveal practices of risk management in Kenyan software projects and the perceived causes for lack of formal risk management practice.

The different categories of primary data collection methods usually used include; laboratory measurements, field observations, archives/collections, questionnaires and interviews [22]. In this study, we opted for interviews using questionnaires as suitable method for obtaining the required data. The questionnaires provide a better way of gathering and recording data while interviews aids to obtain detailed information about personal feelings, perceptions and opinions regarding the topic. It also allows more detailed questions to be asked, yields a high response rate and at the same time, respondents' own words are recorded, ambiguities are clarified and incomplete answers followed up, thus, enabling clarification which gives precise meaning of the asked questions. The questionnaires used in this study were semi-structured and open-ended.

Prior to determining the sample for the survey, a pilot study of 13 software developers was conducted. Valuable information was acquired in the study and the questionnaire was adjusted accordingly. Developers were target at first (pilot study) because we supposed they are the people who interact most with both the clients (users) and the management. Therefore, they had vital information for this study.

The main survey involved 62 software developers and 15 IT project managers from 30 software development firms of varying sizes (small, medium and large firms). The number of IT managers does not match the number of firms surveyed because most IT managers were too busy to participate but some requested their senior developers to participate. Another reason is that, Kenyan software development industry is still young though rapidly growing and the position of project manager is not well pronounced; especially in small and other medium organizations. This is the reason we involved a number of software developers.

The sample was randomly selected from a telephone directory published by Telkom Kenya. Earlier sample targeted 35 IT managers but even after assuring them of confidentiality, 5 declined for the fear of losing their vital information and other reasons which we were not told. We then visited the firms where the targeted persons were requested to participate in the study. All participants were assured that all personal respondents would remain strictly confidential. Then the survey was administered by interviewing participants using questionnaires. The research was administered over a period of two months between January 2009 and February 2009 to the selected software development firms in Nairobi, Kenya.

Nairobi, the capital city of Kenya was purposively selected for the study because virtually all organizations are based in the main capital city where they perform software development activities than at their branches in other locations of the country. Most of the branches only serve as marketing offices. This supposition provided a stronger foundation for achieving a higher response rate.

With the framework's objective in our minds, we obtained facts about Kenyan state of risk management practice, analyzed the findings without comparing results from the two groups (IT project managers and software developers) or based on the size of the organization. Instead cumulative results are given.

Another consideration during analysis was, in the case where two persons from the same organization were interviewed, we became extremely cautious during data tabulation. For example based on policies, only one entry

was considered. Other subjective comments were analyzed cumulatively.

4. Study Findings

Results from the pilot study showed that majority of software developers are not aware of the existence of formal risk management practices. While few development firms with IT policies fail to address risk management practices, majority, do not have policies to govern their processes. Majority of the developers did not understand this area of risk management as most of them based their explanations on software testing.

The main study had five major parts. The first part of the questionnaire (study), "A: Introduction", looked at the background information concerning the interviewees. The study showed that majority of software developers do not specialize in any particular area of development as they develop hospitals' (Health) management systems, financial systems among others.

In second part, "B: Risk Management", the state of risk management practice was explored within the organizations studied. The study revealed that methods used in software projects' risk management included structured or systematic approach (3%), semi-structured approach (10%) and *ad hoc* approach (87%) as shown in Figure 2.

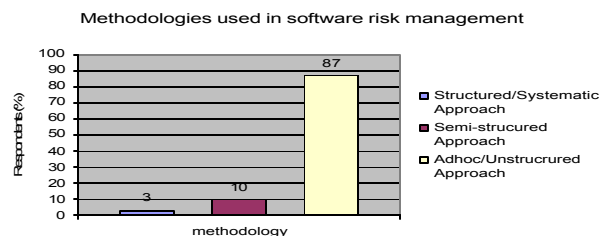


Fig. 2 Methodology used

It was observed that the organizations represented by 3% use structured approach. Greater proportions (87%) do not identify risks and use unstructured risk management methodologies and only 13% of IT managers and developers explicitly manage risks, meaning they have an established structure for risk management. It was noted that some of those who identify and assess risks using structured or semi-structured approaches are ISO certified and complying with ISO standards.

The results further revealed that the same proportion of IT managers that is; 13% documented risk data while majority of them (87%) did not. Greater percentage of

interviewees totally agreed that unstructured methodology is not proper and that risk management is very important area.

Risk communication is another area which is poorly done in Kenya. Most developers believed that, it is the work of the project manager to deal with these risks. Some developers reported risks to their managers but they did not get any feedback on whether they were dealt with or not.

The third part, "C: Awareness", we investigated whether practitioners had some knowledge about risk management practices. Respondents were asked whether they had prior knowledge on risk management techniques (process and tools) as illustrated in Figure 3 and whether they agreed that the formal risk management is important.

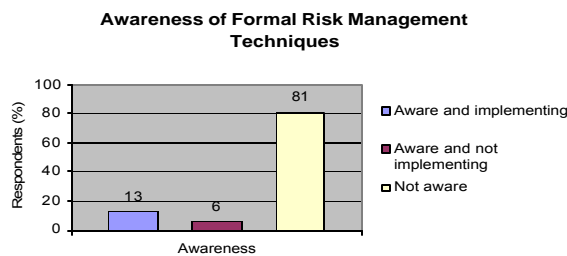


Fig. 3 Awareness of formal techniques

While only 13% of the software project managers and developers had used risk assessment tool before and are still using it (which include: Decision Support System (DSS) risk management support work tools and COBIT® risk management tools), majority (93%) totally agreed that use of systematic risk assessment approaches and tools is important. Software project managers who use DSS risk management support work tools noted that the tool was complex. The tool is also integrated with software development activities and therefore, it cannot capture all the project's risks. COBIT's® risk management tools which include CISA and ISACA does not directly address software development risks but deals with IT governance which are mostly management risks. For example, the risks of implementing and deploying software in an organization.

Majority, 81% of IT managers and developers who use unstructured methodologies lacked awareness of the existence of formal risk management approaches. We took time explaining the benefits of risk management, but greater number of practitioners lacked motivation for risk management activity. Developers believed risk management practice should be initiated by the management. They also believed that, it consumes much

of their time, adds more work and may not bring much difference since projects are usually different in terms of the users, requirements and environment.

The study showed majority of software developers have very little or no knowledge in this area. Most of them revealed that risk management topic was not taught in their programs, and if it was, then it was very shallow. A few claimed that it was mentioned under wide topic of project management.

The fourth part, "D: IT Policies", explored IT policies within the organizations studied, to determine whether they have measures put in place for risk management. This part showed that 80% of the studied firms did not have IT policies as shown in Figure 4. Two firms with policies did not explicitly address risk management in software projects. Therefore, majority of the firms had no measures put in place to address this issue.

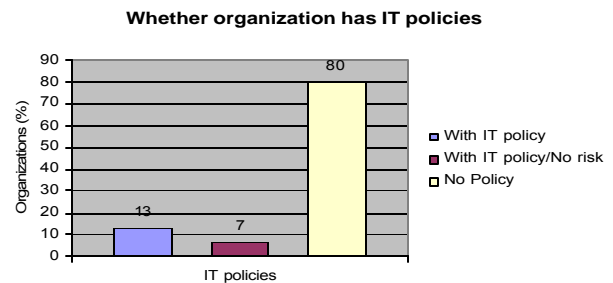


Fig. 4 IT policies

Finally, in the fifth part "E: General recommendations from the interviewee", respondents were given an opportunity to give general remarks concerning the study/research.

General recommendations from the study include:

- The need for clear policies and framework to guide in the implementation of formal risk management techniques
- Importance of creating awareness and conducting proper training in the area of risks both in institutions of learning and software industry
- Establishing proper measures to address various problems.
- Risk management should be implemented and made part of software development culture.

5. Discussion

This study found out that software problems emanate from two major areas; Software Engineering Institutions and Software Development Industry. Software engineering institutions fails to accentuate software risk management

as a course [23] hence graduating students lacking proper basic knowledge of risk management. Another observation made in this study was that software projects fail to meet desired requirements because the applied risk management approaches tend to be *ad hoc*, unstructured, and undocumented. This means, many organizations are not consistently applying existing approaches for software development excellence [24] [25]. Majority of software development firms lack proper communication of risks and in others it does not exist. This has led to ever recurring problems, compromising the quality of software products.

Despite numerous problems experienced by Kenyan software managers and developers, perceived root for these problems include; lack of awareness, undefined policies, poor culture, motivation and lack of proper training in the area of risk management. These causes are discussed as follows:

5.1 Awareness

Kenyan practitioners poorly understand the area of risk management and have very bad attitude towards it. "This shows that, it is not lack of suitable models to help us structure our thinking in this domain; but the real problem is insufficient awareness of current best practices and published standards in software development, management and quality" [26]. Wiegers's sampling report showed that only 10% of the sampled audience members had access to the widely known IEEE Software Engineering Standards. The rest were not aware of their existence [26]. Our study revealed that 81% of participants were not aware of proper risk management practices. This gives us reason to conclude and rate awareness as the main cause for lack of proper risk management practices in Kenyan software projects as illustrated in the model, Figure 5.

Creating awareness will enlighten software development community to employ better practices and eliminate poor attitude towards risk management [27]. Awareness will help practitioners acquire effective skills to deal with various groups of people as well as application of various supporting techniques and tools.

5.2 Curriculum and Training

Ideally, project team members will be able to adapt risk management practices to their particular project environment if team members are well trained [28]. Educating on risk management tools and techniques to software engineering students and developers is important [23] [29]. This is because inadequate training of practitioners and managers in these established practices is a big impediment to the success of software projects [30].

Good education in risk management provides skills and methods for dealing with these problems.

This study shows that majority of Kenyan practitioners lack proper knowledge right from their basic education. There is great need to consider teaching and examining risk management in students' software projects. Educators should incorporate a solid foundation of software best practices, along with guidance about how to put them into action, and into their curricula. Again, there is need to train practitioners to be able to adopt good practices in the area of risk management.

5.3 Policies and Standards

IT Managers and Practitioners may have proper knowledge on risk management. However, this practice may be overlooked if it is not addressed by departmental or organizations software engineering policies. Just as it should be integrated to the curricula so as to be emphasized as a course, risk management process will only be accorded the desired attention when it is incorporated to the policies. This will make stakeholders to take this practice seriously and help eliminate the syndrome "shoot the messenger" [29] [31], where reports on risks in projects are not welcomed as they are perceived as failures.

5.4 Culture

Policies can only be beneficial if they are adhered to and turned to a culture. The culture of an organization is a critical success factor in its process improvement efforts. "Culture is a set of shared values and principles that guide the behaviors, activities, priorities, and decisions of a group of people working in the same area" [32].

Bad cultures in software development industry often lead to problems such as communication breakdown. This is the culture that is being criticized by Boehm and DeMarco [30] saying that "our culture has evolved such that owning up to risks is often confused with defeatism". For example, a project team member may be aware of certain type of a risk but is not communicated because there is a tendency to 'shoot the messenger' often discouraging people from bringing imminent problems to the attention of management.

Turning risk management in software development industry as a culture induces changes that will increase the group's effectiveness and its probability of survival. Wiegers [32] elaborates how software engineering culture which they created in a small software group in a very large corporation improved their effectiveness as software engineers, the relationship and reputation they had with

their customers, their level of teamwork, and the enjoyment they obtained from going to work every day.

This is the main reason Hall [2], Dorofee et al [6], McManus [7], Yardley [8], Boehm [14], IEEE [15], Carr et al [16], and Williams [17] proposed a continuous risk monitoring process which starts from project inception to completion. The repetitive, continuous practice creates a culture that will help practitioners to obtain better assessment accuracy throughout the project life cycle and achieve lifelong success in risk management. Finally, “shoot the messenger” culture should be avoided.

5.5 Motivation

Practitioners may have acquired proper skills and educators as well may have produced knowledgeable students to apply risk management. However, lack of motivation disappoints the acquired knowledge and skills. For example, this study revealed that lack of risk communication is one of the major facilitator of various problems faced in software projects. This basically, is due to the natural tendency of software developers withholding technical information. This may be because of fear, poor attitude and the fact that software developers believe information is a source of power [28] and can help to them survive better in the industry, hence, keeping information to themselves.

Wieggers [25], Walsh and Schneider [33], Schmid and Adams [34] and Barriff [35] believe that if these developers are rewarded for applying better ways of practice and held accountable for sharing knowledge, then lack of motivation can be overcome. Some kind of public praising and commendation seems to help build the spirit of striving for excellence that we all want in our teams [32].

6. Proposed Adoption Framework

Our recommended framework consists of the following:

- 1). A mapping of identified causes for lack of formal software risk management techniques and the implementation of the existing formal risk management processes as shown in Figure 5.
- 2). An approach for tackling software project risks based on continuous improvement and establishment of new measures whenever the implemented measure(s) at any particular time appear ineffective or when other improved techniques become available.
- 3) An illustration of how the approach can be employed in a software project, enhancing mechanism for practitioners and upcoming practitioners. We selected a model for illustration (Figure 6 IEEE risk management process) from the surveyed literature.

Below, discusses how to implement the adoption framework:

6.1 Creating Awareness

In order to create awareness;

- The project managers together with the educators should constantly research and do a review of existing or improved processes which can be used to enhance development processes. Awareness should be taken as a continuous activity.
- Both practitioners and educators can then create awareness through the following ways:
 - Software Engineering Seminars & conferences
 - Journal publications
 - Through software engineering institutions adopting risk management in their curricula
 - Encouraging practitioners and students to practice continuous learning in order to improve software development processes.

6.2 Developing Curricula

- The team concerned with curricula in software engineering institutions should revise their courses and include software risk management as a separate and an examinable course; not as a topic in another course.
- Curricula team members should review the curricula to meet the market demands and include new, innovative and acceptable risk practices
- The team should assess and make sure that the curriculum is well implemented by devising a method of evaluating educators through the learners.
- The curricula team should work together with the educators to revise and improve the curricula
- The curricula team should get feedback about the course directly from the learners or educators

6.3 Training

Training is a continuous exercise that must be conducted in both software industry and software engineering institutions.

1). In Software industry,

- The person concerned with staff training should develop staff training schedule. If the position is non-existent then it should be established in a suitable manner. For example, constituting a

committee or a panel to recruit the training personnel.

- The recruited member, panel or committee members should first be trained on this area.
- Then, they can start selecting the members of staff to be trained
- They can perform the training themselves or invite or employ experts in software risk management
- Motivate the trained staff. For example, awarding them certificates of completion or participation.

2). Institution of Software Engineering should,

- Employ qualified staff in this area of software engineering
- Ensure that the educators cover both breadth and depth of the course as detailed on the curricula
- Examine and assess risk management in students' software projects
- Evaluate the outcome; improve on teaching methods and approaches and review the curricula if need be.
- Ensure that the educators motivate students using various experiments, case studies and awarding good marks to deserving students or scholarships to undertake a research in risk management.

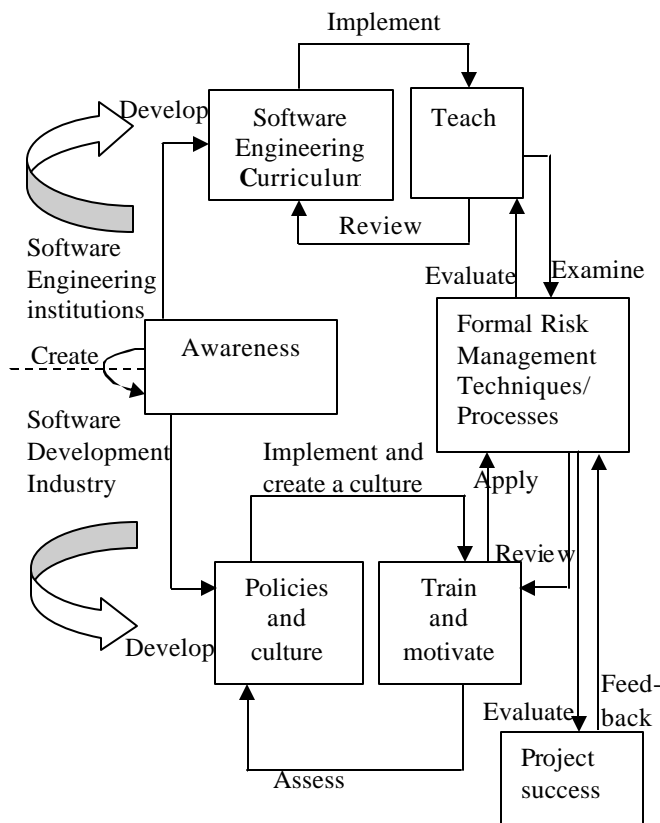


Fig. 5 Proposed risk management adoption framework.

6.4 Creating Policies and Standards

- The personnel in-charge of policy creation should incorporate formal ways of managing software development risks. If there are no personnel, panel or committee looking after policy issues, then it should be constituted. Policy makers should ensure that:
 - Software development policies are developed or existing policies revised to meet the demands
 - The developed policies properly address risk management as well as other practices in software engineering.
 - Acceptable standards are well stated.
 - Make sure that these standards are properly implemented
 - Make sure that continuous risk assessment approach is adopted
 - Policies and standards are revised when need arises
 - Policies address staff training in the area of risk management.
 - Standards address risk documentation and risk improvements
 - Policies and standards enable assignment of risks to members of staff
 - Policies address risk communication by all members

6.5 Developing Culture

In order to develop risk management culture,

- The management team should encourage application of formal risk management techniques to all types of software projects (both small and big projects).
- Every member in development team should be given an opportunity to freely share their perceived risks in the projects
- Selected techniques must support continuous risk management; from project inception to the end.
- Every project's outcome must be evaluated and the techniques are enhanced
- Communicated risks must be looked into without discrimination.

6.6 Motivating Practitioners

To motivate practitioners,

- The management should recognize hardworking and supportive members without any discrimination or biasness. (By giving them good incentives, promotions, letters congratulating them for job well done, opportunities to participate in fully sponsored workshops, seminars and training)

- All the members should be given equal opportunities and attention when reporting risks
- Encouraging knowledge sharing by allowing members hold seminars and workshops in the organization. These activities must be well supported by the management.

6.7 Formal Risk Management Process

This paper proposes the implementation of existing and standardized risk management process. Without proposing a model or process to practitioners and upcoming developers, this paper gives an illustration of formal risk management process using IEEE Standard 1540-2001 risk management process, Figure 6.

Practitioners need to consider a number of factors that influence the successful implementation of existing risk management practices. These factors include;

- Awareness of past risks
- Creating a commitment to risk awareness and action
- Using design approaches that minimize risks
- Ensuring that risk management is forward-looking [36]

Apart from considering the above factors, project managers should ensure that the selected formal methodology (processes and tools) support the following features:

- It should be well structured and standardized
- It should be proactive.
- It should be easy to understand and implement
- It should integrate well with the development processes
- It should be efficient, and not too costly.

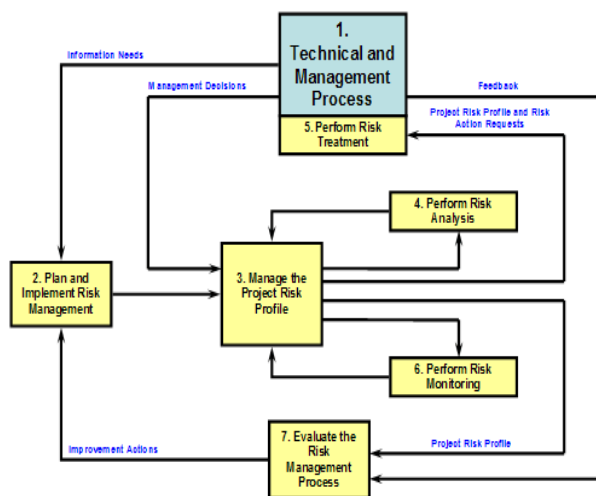


Fig. 6 IEEE risk management process. Source: IEEE Standard 1540:2001.

7. Conclusions

Formal risk management process provides multiple benefits to both the project team and the entire development organization and these benefits can only be achieved if practitioners become aware of these techniques. Therefore, creating policies and awareness of the formal risk management techniques and tools is crucial in this field. Also, educating young software engineers will not only help improve software projects but can lead to innovation of other better ways of handling risks in the industry. This paper notes that only motivated persons will be able to employ proper practices, go to an extra mile; hence deliver more.

Acknowledgments

The authors would like to thank the German Academic Exchange Service (DAAD) for funding this research.

References

- [1] B. J. Williams, J. Carver, R. Vaughn, "Change Risk Assessment: Understanding Risks Involved in Changing Software Requirements", SERP'06-The International Conference on Software Engineering Research and Practice, 2006, pp 2.
- [2] E.M. Hall, *Managing Risk: Methods for Software Systems Development*, Addison-Wesley, 1997
- [3] R.P. Higuera, D.P. Gluch, A.J. Dofee, R.L. Murphy, J.A. Walker, R.C. Williams, "An Introduction to Team Risk Management", SEI Special report CMU/SEI—94-SR-01, Carnegie Mellon University, Pittsburgh PA, May 1994.
- [4] Hughes and Cotterell, *Software project Management*, 3rd Edition, The University of Michigan, McGraw-Hill, ISBN 9780077098346, 2002.
- [5] D.C. Hall, "Development program risk management: a case study". In *INCOSE Proceedings of a Symposium on Risk Management*, 2001, pp 3.
- [6] A. Dorofee, J. Walker, C. Alberts, R.Higuera, R. Murphy, and R. Williams, "Continuous Risk Management Guidebook", Software Engineering Institute, Carnegie Mellon University, USA. 1996.
- [7] J. McManus, "Risk Management in Software Development Projects", Elsevier Butterworth-Heinemann, ISBN 0750658673, 2004.
- [8] D. Yardley, *Successful IT Project Delivery*, 1st Edition, Addison-Wesley, UK, 2002.
- [9] M. H. Roberts "Risk Management in Software Projects". A straightforward approach to project risk management, 2008. [Available online at: www.hraconsulting-ltd.co.uk]

- [10] S. C. Misra, K. Vinod, and K. Uma, "Different techniques for risk management in software engineering": A review ASAC, Banff, Alberta, 2006. [Available online: <http://luxor.acadiau.ca/library/ASAC/v27/content/authors/m/misra,%20subhas/DIFFERENT%20TECHNIQUES.pdf>]
- [11] D. Karlström, Integrating Management and Engineering Processes in Software Product Development, E-huset tryckeri Lund, Sweden, 2004.
- [12] I. Sommerville, Software Engineering, 7th edition, Addison Wesley, Harlow, England, ISBN-13: 978-0321210265, 2004.
- [13] J. Kontio, Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation, Thesis, Suomen Laatu keskus, Finland, 2001.
- [14] B.W. Boehm, "Software risk management principles and practices", IEEE Software vol 8 no.1, 1991, pp.32-41.
- [15] IEEE Standard 1540-2001, "IEEE Standard for Software Life Cycle Processes — Risk Management", Institute of Electrical and Electronics Engineers, Inc New York, NY, ISBN 0-7381-2834-1, March 2001, pp.17.
- [16] J. Carr, L. Konda, Mornarch, F. Ulrich, and F. Walker, "Taxonomy-Based Risk Identification". (CMU/SEI-93-TR-6. ESC-TR-93-183), Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. 1993.
- [17] R.C Williams, G.J. Pandelios and S.G. Behrens, "Software Risk Evaluation (SRE)" (CMU/SEI-99-TR-029. ESC-TR-99-029). Software Engineering Institute, Carnegie Mellon University, 1999.
- [18] R.P. Higuera, Y.Y. Haimes, "Software Risk Management", SEI report CMU/SEI—96-TR-012, Carnegie Mellon University, Pittsburgh PA. June 1996.
- [19] L.H. Rosenberg, T. Hammer, A. Gallo, "Continuous Risk Management at NASA", Applied Software Measurement/Software Management Conference, San Jose, CA. February 1999.
- [20] F.J. Sisti, S. Joseph, "Software Risk Evaluation Method". In SEI report CMU/SEI-94-TR-19, Carnegie Mellon University, Pittsburgh PA. December 1994
- [21] R.L. Van Scoy, "Software Development Risk: Opportunity, Not problem". In SEI report CMU/SEI-92-TR-30, Carnegie Mellon University, Pittsburgh PA, September 1992
- [22] J. A. Sharp. and K. Howard, The Management of a Student Research Project, 2nd Ed, Aldershot: Gower, 1996.
- [23] B. Boehm, D. Port, Educating software engineering students to manage risk. In: Proceedings of the 23rd International Conference on Software Engineering, 2001, pp.591-600.
- [24] A. Tiwana and M. Keil, "The one-minute risk assessment tool". Communications of the ACM Vol. 47, No. 11, November 2004, pp. 73-77.
- [25] E. K. Wiegers, "Practical Project Initiation: A hand book with Tools. Know your Enemy". (Reprinted), Microsoft Press, 2007, pp. 77- 88. [Available online at: www.projectinitiation.com/bookinfo/index.html]
- [26] E.K. Wiegers, "Read My Lips: No New Models!," IEEE Software, vol. 15, issue 5, September/October 1998, pp. 2&5 Reprinted and available at: [http://www.processimpact.com/articles/no_new_models.pdf]
- [27] J. Miler, J. G'orski, "Software support for collaborative risk management", Pro. Of 8th International Conference on Advanced Computer Systems, Mielno, Poland, October 17-19, 2001, pp. 8.
- [28] Y. H. Kwak and J. Stoddard, "Project risk management: lessons learned from Software development environment", Elsevier science, 2004. pp.918-920 [Available online: www.elsevier.com/locate/technovation]
- [29] D. Port, B. Boehm, "Introducing risk management techniques within project based software engineering courses", Computer science education, Volume 12, issue 1&2 2002, pp 37-55.
- [30] B.W Boehm, T. DeMarco, "Software risk management", IEEE Software vol.14, no.3, 1997, pp.17-19.
- [31] J. O'callaghan, project risk management, ISACA, 2006.
- [32] E.K. Wiegers, "Creating a Software Engineering Culture," Software Development, vol. 2, no. 7 July 2004, pp.4-8. Reprinted and available at: [<http://www.processimpact.com/articles/culture.pdf>]
- [33] K.R. Walsh, & H. Schneider, "The role of motivation and risk behavior in software development success", Information Research, <http://InformationR.net/ir/7-3/paper129.html>, Vol7 No 3, 2002.
- [34] B. Schmid, and J. Adams, "Motivation in Project Management: The Project Manager's Perspective", Project Management Journal, vol.39 no.2, 2008, pp 60-71.
- [35] L.E. Barriff, "Human factors in Mission assurance", Aerospace conference, 2003, proceedings 2003 IEEE, Volume8, pp. 8_3769-8_3775.

[36] Data and Analysis Center for Software (DACS),
“Software Acquisition Gold Practice Formal
Risk Management”. [Available online at:
<http://goldpractice.thedacs.com/practices/frm/>]

Noela Jemutai Kipyegen holds Master of Science in Software Engineering, awarded by Jomo Kenyatta University of Agriculture and Technology, Kenya in 2010. Research funded by German Academic Exchange Service (DAAD). Bachelor of Science in Computer Science, awarded by Egerton University, Kenya in 2006. Currently, she is an Assistant Lecturer and Researcher at the Department of Computer Science, Egerton University, Kenya. Research interests include Software Engineering Project Management, ICT for Development, looking at Human Computer Interaction (HCI).

Waweru Mwangi is a PhD holder in Information Engineering awarded by Hokkaido University, Japan. He is a senior Lecturer at Jomo Kenyatta University of Agriculture and Technology, Kenya.

Stephen Kimani holds PhD in HCI awarded by La Sapienza University, Italy. He is a senior Lecturer at Jomo Kenyatta University of Agriculture and Technology, Kenya.