

Significance of Healthy Organizational Culture for Superior Risk Management during Software Development

Charu Verma¹, Saad Ali Amin²

*Faculty of Informatics
British University in Dubai, Knowledge Village
Dubai, U.A.E*

¹charudeepi@yahoo.com

²csx188@coventry.ac.uk

Abstract

This paper proposes a support framework which widens the area of software risk management by exploring numerous cultural factors, which may lead to variety of insignificant risks during different stages of software development and subsequently turn into major causes of software failures. In brief, this study proposes step-by-step guidelines to deal with the basic problem area that is insufficiently addressed in today's "state of art" era: The unhealthy work culture effecting risk management during software development. This paper is an outcome of research conducted on "The Effects of Organizational Culture on Risk Management during Software Development" for which several research methodologies have been adopted in order to explore the area of study. The research analysis shows that the relationship between work culture and risk management was consistent across a number of measures adopted by various organizations, which clearly results in either a successful or a failed software project. Hence, based on the findings of the study, this paper identifies various cultural aids in the form of suggested guidelines for effective risk management, and illustrates the emergence of healthier risk management culture during software development in an organization.

Keywords: Organizational Culture; Risk Management; Software Development; Project Management; Risk Culture

1. Introduction

Given the rise in technology and increased variance in software tools, it is evident that each software development project is unique. As a result, software development projects have an unfortunate track-record of cost and schedule overruns along with quality and usability problems. Therefore, it is dire for a development team to preemptively prepare for these delays or mishaps. It becomes the responsibility of the management, followed by all the stake holders to induce a healthy environment, which encourages smooth operations during critical and complex development stages. The focus of this paper is to provide an understanding of the key barriers to effective software project risk management and how to overcome them by making it an element of the work culture.

Further sections provide information about various areas of study considered for the research in order to develop the guidelines. Subsequently, it discusses the research methodology adopted in order to research the areas considered. Finally, the suggested guidelines and its integration with Software Development Life Cycle have been illustrated.

The Figure.1 hereunder outlines the areas covered in the research and the approach adopted in order to develop the suggested guidelines. The aspects mentioned in the figure have been taken into account while developing the solutions framework.

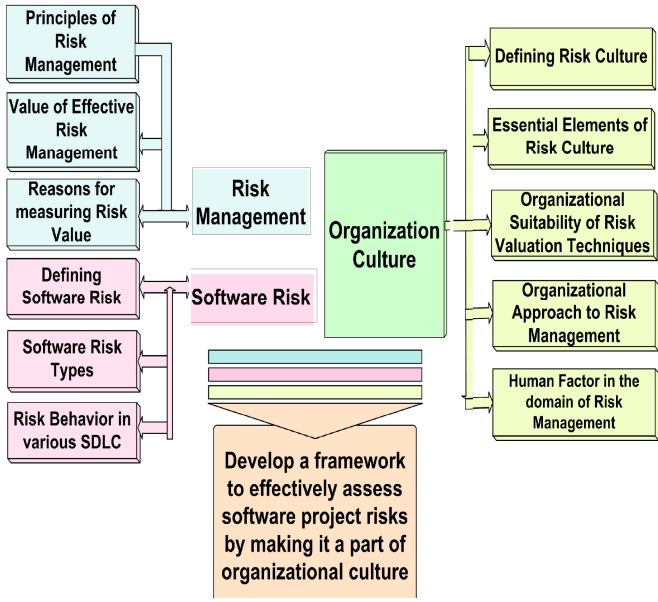


Fig.1 Approach to framework development

1.1 Risk and Organization Culture

As quoted by Henry (2004), “A project team, like a football team, is a cultural entity and the first step towards shaping the culture of a software project is to understand the culture of your organization”. If software engineers have little confidence in management, struggle with computer aided software engineering (CASE) tools, have no software process to follow, or see schedules as pure fiction, a project attempting to be strong in these areas faces an uphill struggle.

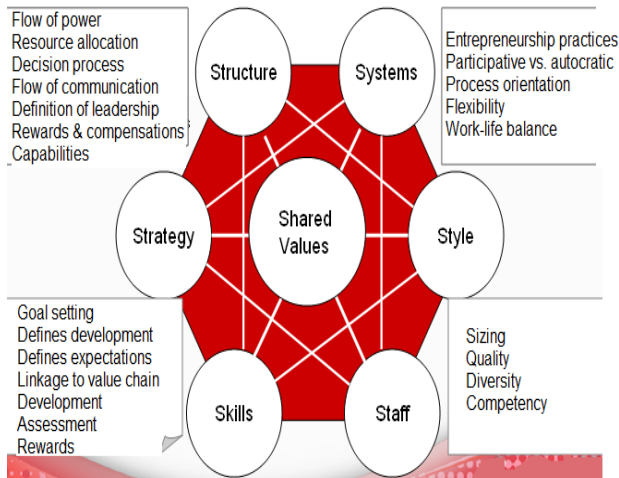


Fig.2 Seven determinants of culture in an organization (Malia, 2006)

cultural equilibrium with the 7 determinants of culture as mentioned in the very comprehensive Figure.2. When paradigms shift, everything is reduced to zero and a balanced state of culture becomes imbalanced”. Through diagnosis, we need to test each vertical ‘S’ (shared values, systems, structure, skills, style, staff and strategy) and its impact on the behaviors required for the new culture by performing interventions accordingly to arrive at the new equilibrium. In order for an organization to build a healthy culture, it does not necessarily need to train the staff in the above mentioned seven determinants; rather it essentially needs to implement a flourishing environment where its employees are able to perform for the benefit of the project as well as their personal development. Therefore the correlation between good culture and better risk management during SDLC has been taken into consideration. SDLC is an acronym for System Development Life Cycle which forms the basis for Software Project Management.

According to Pritchard (2007), Senior Consultant, Cutter Consortium, in order to know what is missing from an organization's risk culture, one must first determine which elements are required. If, for example, an organization has the financial resources to endure nearly any risk, then setting up cost thresholds should be the decisive factor. An organizations' risk culture is composed of consistent terminology, thresholds, triggers, mandatory practices, and controls. Within these elements, a unique set or practices determines the treatment of these elements.

Risk culture flourishes on clear definitions of risk terminology. In building a risk culture, it is critical to share the glossary with all decision makers and reach common terms of usage agreement by defining them in a way that makes sense to all and is simple understand. This avoids differences and deviation while implementing risk mitigation plans. Some of the key terms as described by Pandian (2007) that need definition for clear understanding and usage, are summarized in the Fig.3



Fig.3 Risk terminology summarized

Malia (2006) in his speech elucidates that “a successful organization in a given paradigm is always in a state of

Tolerances refer to limits of organizational behavior. This is the boundary an organization jointly will not cross. For instance, a big company not accepting very small projects.

Thresholds are probably the most universal and easily developed elements of a risk culture. It decides the extent to which an organization will go during odd times. Organizations generally draw these limits on costs, schedules, employee conduct, cultural knowledge, community participation, and a herd of other concerns. Strangely enough, these thresholds are rarely announced, they are instead passed on from one employee to another as part of an organization's inherent tradition. New members of an organization learn culture of risk handling by virtue of unpleasant experiences. On encountering a trouble and being denounced for it, new employees are introduced into the culture. Absence of documented thresholds may bring in two concerns: difficulty in communication about them and lack of consistency in values inherent in the form of thresholds. For instance, maximum amount of money a company is capable of investing in a project.

Triggers are the caution signs that point to a threshold about to be violated. For instance, project cost during development is about to reach the threshold amount, which was previously set while planning the functionalities.

Mandatory process refer to the “must do” list of an organization. It is important for some critical situations, where individual style of dealing may not be enough. They also make sure that communication takes place on a regular basis among involved parties in case of experiencing any risks. Good example can be – a software developer producing a daily report of his completed task list or studying previous projects-reports of similar nature before starting a new one, so that he/she can identify comparisons and deal accordingly with project complexity.

In good practice organizations, internal risk management practice many times involves measuring the performance. These practices being in place ensure the existence of a consistent behavior, for instance providing management with timely updates or documenting the potential problems. On the contrary, a backup plan of risk responses will act in the absence of these practices.

Risk controls act as "thermostats" of the risk management procedure, which can be set for problem detection and forgotten, provided there are some pre-existing thresholds in the process. This helps in getting used to a consistent practice of “how to deal with risks”.

In an organization where risk controls run successfully, staff is capable of defining risks effectively, describing organizational risk strategies efficiently, hence manage to handle risks proficiently.

2. Research Methodology

With a view to attain an in-depth understanding of software project risks and how they are handled or perceived because of good/poor organization culture, it was necessary to espouse a variety of research methods. This section summarizes various methods of research on the grounds of why they were adopted and an explanation of how each method was developed. There were four main types of research carried out which are outlined in the table below:

Method	Grounds for choosing the method
Desk Research	Involved consistent study of existing work (using books, online articles-journals, magazines). For the phase two it was used to develop the issues found in phase one. Finally it was used for developing and verifying the recommended framework for phase three.
Interviews	The interviews were conducted with top managers at four big organizations, who closely encountered software project risk management issues, resulting from poor/good organization culture.
Case Studies	Five case studies were employed, some in-depth, while others were used with a view to validate a point. They provided an industry-wide perception of the software risk issues which were concerned with organization culture whose solutions which are currently popular.
Surveys	The survey referred to is a compilation of many surveys: Standish Group; AFSN Organization; ITcortex; Repario and Loon. Questions are compiled after researching on various surveys available on “organization culture effecting software development”. Also it was supported by the thesis supervisor who himself had been a part of many software development projects.

Table.1 Description of research methods

The research techniques adopted in this study were mainly qualitative. There are two main reasons for this. Firstly, desk research shows that in spite of enough statistical studies conducted on the types of software project failures, reasons of failures, number of failures and software risk management, there has been very little

discussion on the topic of risk management culture, both by organizations and also collectively (industry wide).

2.1 Framework Foundation

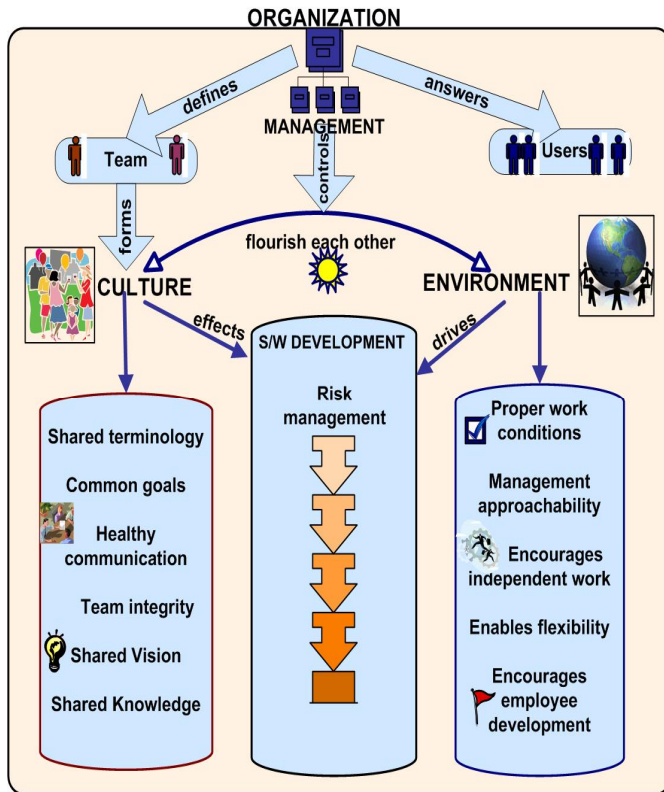


Fig.4 Software risk management - dependence on the organization culture

After a thorough analysis of findings and discussion of the study, key elements of an organization have been identified and presented in the Figure.4, which could impact the risk management process during software development. The point made by Ewusi-Mensah (2003) is very interesting as it touches on three aspects which have been discovered repetitively within his study on software projects with respect to organizational environment - 'management perception, commitment and pressures'.

3. Framework Development

Research analysis leads to the first derivation that *Planning and Design* phases in Software Development and *Identification and Analysis* phases in Risk management decide the fate of a project. Carefully looking at any SDLC of similar projects, the development phase may involve same coding language and same process; however the successful completion of

any software project is highly dependent on how the team understands it and plans it for the tasks to be carried out and risks to be treated. Although there are numerous other factors which trigger risks, however most of work gets completed when the team and the project manager feels confident about taking the project ahead by planning tasks from requirements and identifying risks from the tasks.

The second most important derivation is: *risk management is not just one time function*; it begins with the project planning and goes beyond implementation. The way successful projects suggest testing a piece of code at least 3 to 4 times a day, similarly Risk Management needs to be carried out at least 4 to 5 times a week. According to Standish Group (2007), one of the greatest risks which cause software failures is the sales and marketing people crunching numbers due to sales pressure. However, this study focuses more on the type of projects contracted by a client for specific requirements; therefore this ground has been ruled for its inclusion in the causes of software failure.

According to a global study conducted by IBM (2008), majority of organizational change projects failed in 2008. IBM's "Making Change Work Study" divulges that nearly 60% of projects aimed at achieving business change fail to meet their goals fully, but that the most successful organisations - the Change Masters - have cracked the code and succeeded in 80% of their projects. In sharp contrast, the bottom 20% of the sample, referred to as Change Novices reported a project success rate of only 8%. This data proves that people are reluctant to changes in operations. However an organization needs to identify the reception level of its staff before implementing a change. This also implies that the change is meant for the business, but not the business for the change. Therefore careful study of existing situation (in the organization) is required before implementing any change and same applies for a SDLC. Secondly, as there are numerous methodologies and suggestions by various experts in the field of software project risk valuation, it was obvious that an attempt to gain first hand statistical research would have provided inconsistent measures.

With the collapse of *Toyota Prius* and other large software projects, the issue of corporate governance has started to gain pace recently. This meant that the area of software risk management is still developing, and attempts to find quantitative data were deemed to be outdated or worse still misleading. The survey questionnaire found to be of real value, which has been used as one of the basis for developing a suitable support framework.

3.1 Risk Management Guidelines – Healthier Culture.

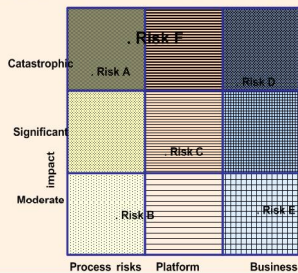
Risk Management Process

Step 1. Identify Risks

- Identify on the basis of complexity.
- Team brain storming – empowers trust.
- Study previous projects.
- During team discussion, include at least 1 person with experience in similar project.
- Identify through out SDLC.
- Spot risks during requirements gathering.
- Assign one “risk tracking” each to a team member.
- Use risk terminology familiar to all.
- Give freedom to team members while risk definition.
- Also spot success factors.

Step 2. Analyze Risks

- Analyze only what is most important at present.
- Analyze tasks which may cause risks.
- Don't spend too much time as time is precious.



Step 3. Plan and Resolve Risks

- Prioritize on the basis of current situation, don't over plan.
- Keep eyes open for time and resources needed, it should not out weigh the benefits.
- Discuss more and more about the issue, during informal meetings.
- Research on similar risk planning if the situation is difficult.
- Employ prototyping – can help foreseeing more risks.
- Take a look at staffing decisions – to be changed or revised.
- Always monitor quality, it should always increase the project quality.
- Focus on 20% of most likelihood risks which cause 80% of problems.

Step 4. Control Risks

- Clarify ownership issues and award for risk handling.
- Reward for risk avoidance.
- Build a work habit where technical lead & developers must be able to state their top 3 risks at every stage of risk identification.
- Use only those risk control measures which simplifies the already difficult development process.

Step 5. Treat Risks

- Make a log of risks solved for future reference and new members.
- Track risks with associated tasks and make sure they don't re appear. This may be of great help on the organizational level.

Fig.5 Superior cultural aspects of risk management process

Cultural forms facilitate how people make sense of their world (Ovaska, 2008). Consequently, this section discusses various steps in order to identify an adaptable risk management culture, depending upon project type, organization type and employee type. It is not necessary that implementing this solution will ensure a definite victory; however it definitely ensures

a project not falling into a trap of disruptive failure, when sometimes it becomes impossible for the team to carry the project ahead for not being able to deal with various insignificant factors. Following support framework aims for eliminating many of these factors, which in later stages of development may cause major risks. As the risk management in general is known to all, the following section will emphasize on the “working style” or the *Cultural Factors*, which help in managing risks during software project development process.

In a setup where the software development process is apt but the risks are not handled properly or a setup with foolproof risk management process with inefficient software development knowledge, it is apparent that the chances of project failure are high. Therefore, a list of suggested risk management guidelines in the Figure.5 has been construed and encouraged to be followed all through the SDLC. Furthermore, a SDLC culture has been explored and discussed, in order for suggested risk management guidelines to be a success. This has been achieved by integrating suggested guidelines into a SDLC. As both are necessary for each other and complement each other.

3.2 Integration of guidelines and SDLC - How a good work culture manages Risks

3.2.1. Project planning and requirements gathering - Identify risks

Following steps, if followed in the first stage of project planning may possibly avoid several risks, right in the very beginning.

- **Freedom of decision making and participation** should be provided to the team in the beginning because it is the team who handles the risks during later stages. This improves the understanding of system functionality in the minds of developers, managers and the client.
- Interview results of four IT managers proves that a Project Manager or the Project team should have a **face to face meeting** with the user in order to get better clarity on requirements, by dividing requirements into one of the following three categories (as in the Figure.6, bottom) and begin with most important features and *only what is in hand*. Better the understanding, easier is to deal with scope changes.

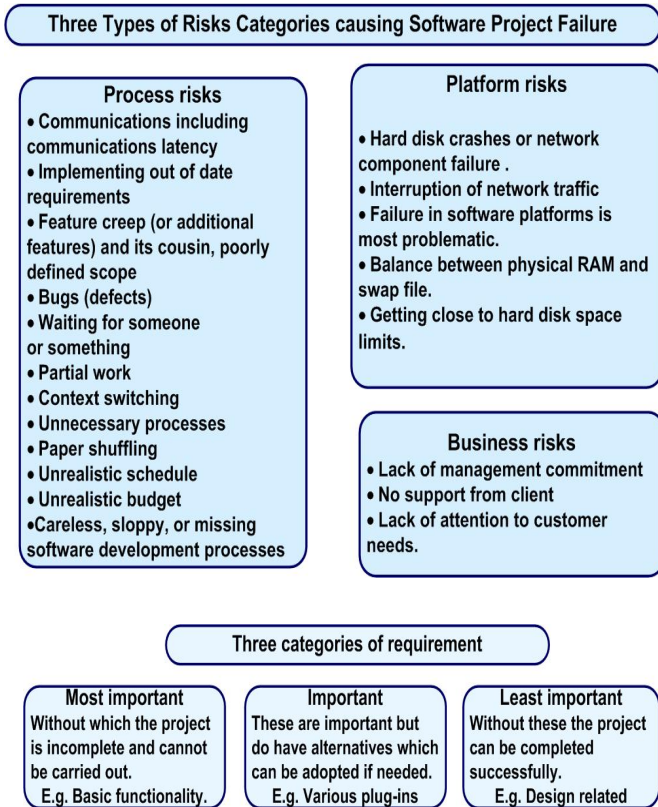


Fig.6 Risks(top) and requirements(bottom) categories summarized

- **Brainstorming requirements** with the team can expose some hidden **threats** or golden **opportunities** that are coming up on the way (IBM and Microsoft). Interviews, team sessions (risk brainstorming) are some common techniques to accomplish this. A helpful method is to put the participants into different teams to identify the project risks, prior to engaging in a group discussion. As suggested by Krigsman (2008), identifying risks in categories (Figure.6, top), will help in their better prioritization.
- **Project value** should be evaluated for complex projects with the help of ROI, IRR, and NPV, which actually measures the **risk value** for the project during requirements gathering stage.
- **Provide examples of the value of cultures**, using various academic and corporate sources (Articles and presentations available from websites) backed up by structured thinking on the subject of risk management by the team.
- **Selection of a SDLC** should not depend on its “name”; rather it should depend on the level of completeness and understanding of requirements.

- The requirements gathering process should be kept simple, keeping in mind a vague idea of how the client visualizes the system.

3.2.2 Project Design – Access and Analyze Risks

- **Brainstorm with the team**, by utilizing their previous experiences, project plans, the company Intranet, and assistance websites, then design for basic functionality with the existing data.
- **Empower team** by providing them an opportunity to express views and suggestions, which again improves the ‘process quality’ and saves time. Also *Design with keen insight* and spend more time than development stage. This will *save time* by encouraging lesser testing.
- **Design a test plan or a list of important system components to be tested** at this stage, so that testing does not drag till completion phase. This further avoids schedule over-run. **Involve software engineers** during architectural design stage, which will help them deal better if requirements change in future.

Understanding the nature of a risk is a prerequisite for a good response. Therefore take some time to have a closer look at individual risks and do not jump to conclusions without knowing what a risk is all about. Risk analysis occurs at different levels. If one wants to understand a risk at a particular level, it is most fruitful to think about the effects that it carries and the triggers that can cause its occurrence. A more detailed analysis of risk treatment during “the analysis stage” has been worked upon in the Figure.8, which shows the order of magnitude of risk effect and how the team can deal with risks by assigning them certain ‘effect category’.

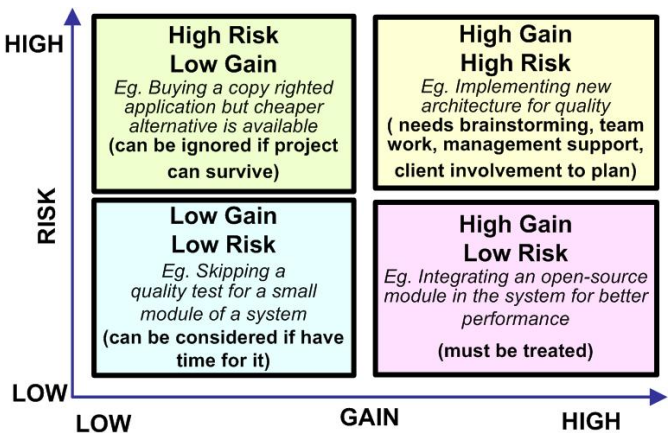


Figure 8: Analyzing Which Risks to Respond

3.2.3 Project Development and Testing -- Handle risks

This is the phase when project visibility improves and all the requirements planned and risks identified take a physical form. The testing phase has been fused with development phase because a good work culture is being discussed throughout the paper, which helps reduce risk occurrences in a SDLC. Furthermore,

- **Sound development principles** should be employed. Say high quality coding standards, organizational training, and training to increase knowledge of the developer.
- It is very important to keep **communication on** about risks being encountered. Frequent meetings should be conducted in order to handle them. **An informal meeting during this phase is the key to success**, as the developers are under a lot of stress of programming. These meetings sometimes turn out to be of great help in cracking the problem.
- **Test with stability and integrity.** Which means the system should not only complete but run smoothly after deployment. Because risks could be more frightening if they appear after completion.
- Make **use of old projects** to check how similar risks were handled as practiced by most of the successful organizations studied in the research.
- **Prioritize risk responses**, keeping in mind that responses should not be handled at the cost of project budget. **Spot opportunities:** Risk management does not mean only avoiding risks but also helps in spotting opportunities for more financial gains.
- **Report to the customer** on completion of every module (if the project is big), or at least number of times during the entire project life (if small). Because there are many project managers or relationship managers (who communicate with the client regarding the project's progress and) who show up only after the project is complete or when it is bound to fail.

3.2.4 Project Implementation – Risk Treatment for Future Reference

With the end of development phase, risk management is not over. The 'good culture' being discussed tries to deal with: Discovering risks for their treatment; Playing with risks for assessment, analysis and control; Preserving risks for future reference.

- **Tracking risks** and their associated tasks can be carried out with the help of the risk register. Incorporating risk tasks into daily routine is the key to implement responses.
- **Maintaining a good risk log** clarifies ownership issues, provides risks descriptions, and enables carrying out some basic analysis regarding causes and effects.

4. Conclusion

The suggested guidelines cover the significance of culture (way of working) during risk management process of software development project. More effort is vital to develop this area. Additional work needs to be carried out on the front end of the cultural model of a software setup (culture audits, dimensions and needs). This involves developing measurable outcomes of the cultural risks and performance methods.

The tools and methodologies employed in order to demonstrate this framework are a sound analysis and compilation of existing tools and techniques from a variety of sources. The main sources of these tools are from the techniques developed and espoused by IBM, Microsoft, Infosys, NASA, ChangeSource private ltd., Agile, various software development life cycles and literature review of various companies adopting risk management techniques in their own unique styles and most importantly the findings of surveys, case studies and interviews.

Therefore risk management involves not merely preventing or controlling risks, but it is a way of widening the area of knowledge of one's own expertise in the task one is performing. Furthermore one should take pleasure in risk handling with a learning attitude. Hence, the risks are not to be scared of, because the synonym for "software project" is "risk undertaking".

References

- [1] AFSN, 2008. AFSN Organizational Change and Community Development: *A Case Study*, internet
- [2] Change Source Pvt. Ltd., (n.d.) *Risk Assessment for Projects*, [online]. Available at: <http://www.change-management-toolbook.com/mod/book/view.php?id=74&chapterid=65> .
- [3] Gotterbarn, Don, 2002. *Reducing Software Failures: Addressing the Ethical Risks of the Software Development Lifecycle*. Australian Journal of Information System Software Engineering Ethics Research Institute. East Tennessee State University, USA, Vol.9, No.2.
- [4] Ewusi-Mensah, Kaweku, 2003. *Software Development Failures* . Cambridge: MIT Press
- [5] Hennessy, Joe, 2009. *ISD Software Risk Identification*, Ver.2, 1-6.
- [6] Henry, Joel (NASA), 2004. *Software Project Management*. 1st ed. Montana: Pearson.
- [7] IBM Corporation, 2008. *IBM Global Study: Majority of Organisational Change Projects Fail*, [online], 15th October. Available at: <http://www.ibm.com/news/hk/en/2008/10/15/s921892u38298o66.html> [Accessed 5 May 2009].
- [8] Malia, Adil, 2006. The Role of HR in Organization Culture Building, FCCI Speech Conference . New Delhi: FCCI.
- [9] Ovaska, Päivi, 2008. A Case Study of Systems Development in Custom IS Organizational Culture. Springer Publication, US.
- [10] Pandian, Ravindranath, 2007. *Applied Software Risk Management*, Auerbach Publications:Newyork,
- [11] Pritchard, Carl, 2007. Creating a risk culture in an IT environment. *Cutter Consortium Press Release*. Vol.2, Noi.1
- [12] Standish Group, 2009. *Standish Group Chaos Report finds increasing project failure*. The Productivity Habit. [Accessed 15 May 2009].
- [13] Standish Group, 1995. *The Standish Group Report*, [internet]. Available at: <http://www.cs.nmt.edu/~cs328/reading/Standish.pdf>