

AN EVALUATION OF SOFTWARE PROJECT RISK MANAGEMENT IN SOUTH AFRICA

B. de Wet¹ & J.K. Visser²

Department of Engineering and Technology Management
Graduate School of Technology Management
University of Pretoria, South Africa
¹brendandw@gmail.com, ²krige.visser@up.ac.za

ABSTRACT

Global research indicates that the success rate of software projects worldwide is currently very low, and has been low for the past few decades. The application of risk management has improved the success rate of software projects in the developed world. This study investigated whether the success rate of software projects in South Africa is also low, and whether risk management might improve these success rates. The results indicate that the average success rate of software projects in South Africa is indeed very low, and that software projects in South Africa often experience the same risks as in the developed world. It was also found that, where risk management is applied, software projects produce better results than software projects with no risk management. The majority of South African software companies use ad hoc internally developed risk management procedures rather than formalised procedures.

OPSOMMING

Die sukses tempo van sagteware ontwikkelingsprojekte oor die wêreld is tans laag en volg die tendens reeds vir 'n aantal dekades. Die toepassing van projek risikobestuur het die sukses tempo vir sagteware projekte wel verbeter in ontwikkelde lande. Hierdie studie het 'n ondersoek gedoen om te bepaal of die sukses tempo van sagteware projekte in Suid Afrika ook laag is, en of die toepassing van risikobestuur vir sagteware projekte die sukses tempo kan verbeter. Die resultate van die studie het aangetoon dat die gemiddelde sukses tempo van sagteware projekte in Suid Afrika wel baie laag is, en dat die projekte dikwels soorgelyke risiko's ervaar as in die ontwikkelde wêreld. Die studie het ook bevind dat sagteware projekte waar risikobestuur toegepas is beter presteer het as projekte sonder risikobestuur. Die studie het ook aangedui dat die meerderheid sagteware maatskappye in Suid Afrika gebruik maak van ad hoc risikobestuur prosedures wat intern ontwikkel is, en dus nie formele prosedures gebruik nie.

¹ The author was enrolled for the MSc (Technology Management) degree in the Department of Engineering and Technology Management, Graduate School of Technology Management, University of Pretoria

² Corresponding author

1. INTRODUCTION

1.1 Software project risk management

The unpredictable and dynamic nature of software development projects poses numerous risks and challenges to organisations. Research conducted by The Standish Group [1] indicates that only a small number (32 per cent) of software projects worldwide are successful. The Standish Group considers a project to be successful if it meets all three of the following requirements:

- Completed on time
- Completed within budget
- Completed with the required functionality

The Standish Group [1] further states that 44 per cent of software projects are challenged (they fail to meet some or all of the above requirements), and 24 per cent of software projects fail outright. Fairly [2] suggests that a great number of software project failures could be avoided if risks were properly assessed and mitigated. Various frameworks and methodologies are currently available to aid in the management of software project risks. However, most of these frameworks and methodologies originated in Western/European countries, and were designed to address risks within their respective environments. Consequently, these frameworks might not be suitable for organisations in South Africa to adopt. There is thus a need to investigate the applicability and effectiveness of software project risk management in South Africa.

The information engineering (IE) approach to information management (IM) was developed in the early 1980s to help organisations to manage data and information in a more cost-effective way. Hogan & Raja [3] define IE as “an integrated, full life-cycle systems development approach with automated tool support which can be useful in assisting information systems managers in imposing a rigorous discipline on the systems development process”. IE can also be viewed as a collection of tasks, activities, techniques, roles, and deliverables.

This study seeks to contribute to the IE body of knowledge by investigating project risk management as a technique or method that ensures adequate deliverables of software projects on time and within budget.

1.2 Objectives of this paper

The research project had the following two main objectives:

- To measure the effectiveness of risk management within the South African context
- To identify the optimal method for managing software project risk in South Africa

To accomplish the identified objectives, this study defined the five research questions outlined in Table 1.

2. LITERATURE

2.1 Background

Much research has been done worldwide on the topic of software risk management. This study reviewed various literature sources, extracting relevant information in order to answer the research questions that could not be suitably answered by using quantitative methods.

Table 1: Research questions

Research question	Type of research	Source
1. Can the high software project failure rate be substantiated and consequently confirm the need for the study?	Quantitative	Questionnaire
2. What is the definition of 'risk' within the South African software context?	Qualitative	Literature review
3. Are the software project risks faced by organisations in the Western/European world the same as in the South African context?	Quantitative	Questionnaire
4. Do organisations with formal risk management procedures outperform those without - i.e., does risk management actually make a difference?	Quantitative	Questionnaire
5. Which software risk identification approach has produced the most success within the South African context?	Quantitative	Questionnaire

Misra et al. [4] state that risk management has been used for many decades in software projects. Earlier software development projects in the 20th century conducted risk management using different ad hoc approaches without following systematic methodologies. However, with the increasing complexity of software development, industries have realised the importance of risk management during the development process. Risk management reduces the uncertainties involved in developing software, and decreases the chances of project failures.

A number of well-known formal software risk management methodologies are briefly discussed in this section. Most of these assess risks during each phase of software development, and follow a disciplined process.

2.2 Boehm's risk management model

Boehm's original risk management process ('the spiral model') [5] was developed in 1989. The spiral model eliminates risks from the early stages of software development, as opposed to taking the chance of encountering project barriers at later stages [4]. Boehm extended his original spiral model using the Theory W (Win-Win) model, which aims to satisfy the objectives and concerns of the stakeholders. Boehm [4] also proposed a risk management framework that aids in identifying the primary sources of risk, and subsequently analyses and resolves them. He presents the risk management plan in the form of a tree diagram (Figure 1).

2.3 The SEI Continuous Risk Management (SEI-CRM) method

The SEI-CRM method (developed by the Software Engineering Institute) is a software engineering practice with processes, methods, and tools for managing risks in software projects. It provides a disciplined environment for proactive decision making to:

- continuously assess what can go wrong (risks)
- determine which risks are important to deal with
- implement strategies to deal with those risks.

Esteves et al. [6] explain how the SEI-CRM method is applied to assess risks continuously in software projects, and to use this for decision making in all phases of a project. Risks are carried forward and dealt with until they are resolved, or until they turn into problems and are handled as such.

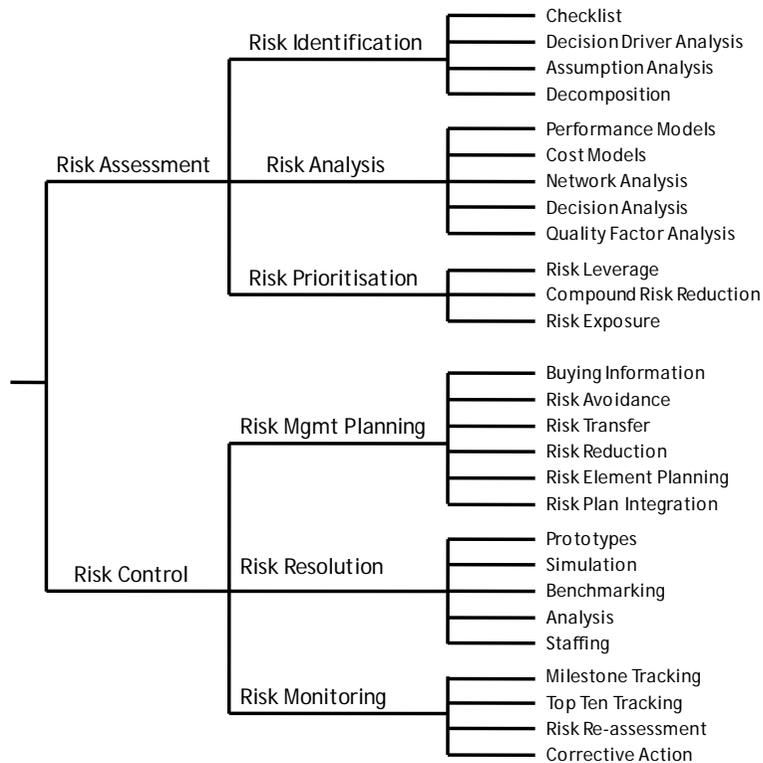


Figure 1: Boehm's model for software risk management [4]

2.4 Kontio's Riskit method [7, 8]

The Riskit method for software engineering risk management, developed by Kontio [7], is one of the most popular and widely-used methodologies worldwide. This method has gone through a number of iterations, and has been improved through the feedback received from various case studies or evaluations carried out in industry organisations. The Riskit method was developed to provide project managers with an explicit and systematic process for risk management, while providing precise definitions of risks and risk effects, as well as tying risks to specific project goals and stakeholders. In doing so, all parties associated with a project can easily understand the risks that they face, as well as their implications. The Riskit method uses a specific ranking technique called the 'Riskit Pareto Ranking Technique', which uses probability and utility loss ranking [9].

2.5 The CMMI methodology

The capability maturity model integration (CMMI) [10] is a model that assists in both the development of software products as well as in their maintenance. The CMMI model provides a structured view of process improvement in all departments or divisions of an organisation. The CMMI products consist of an integration of models from four disciplines (software engineering (SW), systems engineering (SE)), appraisal methods, and training materials). The CMMI can assist an organisation to:

- integrate traditionally separate organisations
- set process improvement goals and priorities
- provide guidance for quality processes
- provide a yardstick for appraising current practices.

Risk management in the CMMI framework is a project management process at maturity level 3. Williams [11] says that all risk management standards suffer from the weakness of promoting 'compliance' with the opinions of a small group of risk practitioners. Williams

feels that the CMMI risk management process, if used as a standard, has the potential to promote and achieve effective risk management.

2.6 Other risk management methodologies

Several other risk management models or methodologies have been proposed, but these are not discussed further in this paper.

- The Softrisk model (Keshlaf [9])
- Hall [12] software risk management process
- Goguen et al. [13] IT Systems risk management methodology
- Thomsett [14] risk management process
- SERIM (Software Engineering Risk Index Management) method of Karolak (See Misra et al. [4] for a discussion)
- ProRisk framework developed by Roy [15]

A discussion of general risk management processes is provided by Samad and Ikram [16].

3. CONCEPTUAL FRAMEWORK FOR THE INVESTIGATION

3.1 Evaluating risk management procedures

Various standards, methods, and processes are available for risk management in projects, and some specific risk management standards and processes are available for risk management in software development projects or software usage and maintenance. Some of the well-known methods and processes were discussed in the previous section. A more detailed discussion of these processes and of the advantages and disadvantages of each is provided by De Wet [17].

In order comprehensively to test the impact of risk management on project success, it was necessary first to evaluate each risk management procedure used in order to determine whether the procedure in question was sufficient. It was thus necessary to measure the impact of risk management by using only sufficient risk management procedures as input. This study evaluated risk management procedures by comparing them with the IEEE Standard 1540-2001 for software risk management [18]. This process was chosen for the study because it is well-known in the software industry, having been applied by many software development organisations for more than a decade, and because it is a standard that was developed by the IEEE, one of the largest of its kind in the world.

3.2 The IEEE Standard 1540 [18]

This standard defines a continuous software risk management process that is applicable to all software-related engineering and management disciplines. The risk management process itself comprises several activities and tasks that function in an iterative manner. The process defines the minimum activities of a risk management process, the risk management information required and captured, and its use in managing risk. The risk management process defined in this standard can be adapted for use at organisational or project level, for different types and sizes of projects, for projects in different life cycle phases, and to support diverse stakeholder perspectives.

It is intended that the standard should be adapted by individual organisations and projects to meet their specific situations and needs. For this reason, this standard does not specify the use of any specific risk management techniques or associated organisational structures for implementing risk management. However, the standard implicitly supports the use of tools and techniques that can make risk management a continuous process. Capturing and communicating risk-related information in electronic form to all parties involved in a project is encouraged. The IEEE standard for risk management is illustrated as a block diagram in Figure 2 below. The purpose of each activity is discussed briefly.

3.2.1 Plan and implement risk management

The purpose of this activity is to establish a software risk management process. Where an organisational risk management process already exists, the software risk management process should be aligned to it.

3.2.2 Manage the project risk profile

The purpose of this activity is to create a consistent, current, and historical view of the risks that are present, along with their treatment, so that the risks can be communicated fully and succinctly to relevant stakeholders.

3.2.3 Perform risk analysis

This activity should be performed continuously throughout the software's life cycle, and should consist of risk identification, risk estimation, and risk evaluation.

3.2.4 Perform risk treatment

This activity should contain the following tasks: selecting risk treatment; risk treatment planning and implementation; and performing risk monitoring.

3.2.5 Evaluate the risk management process

This activity should contain the following tasks: capture risk management information; assess and improve the risk management process; and generate lessons learned.

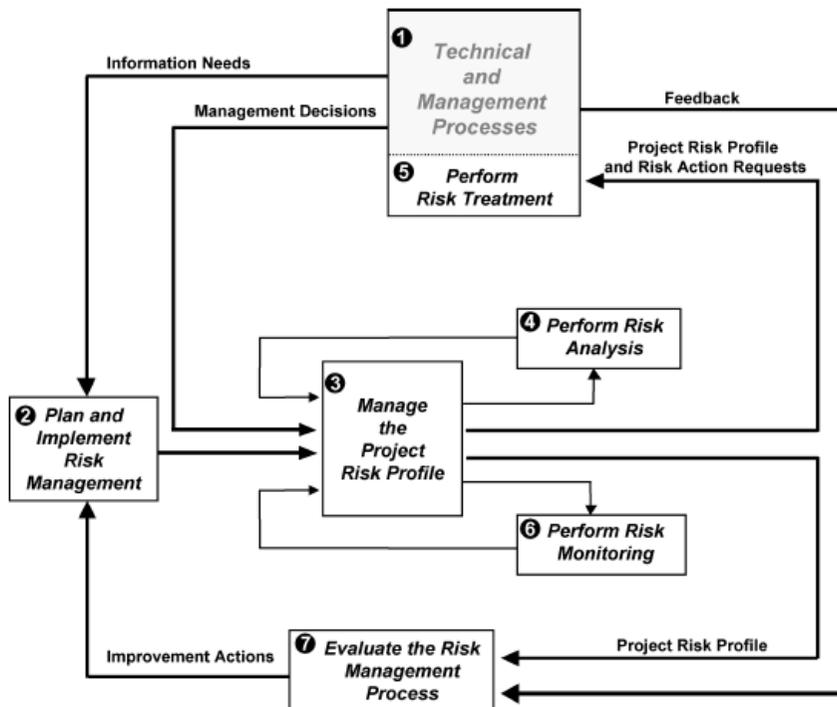


Figure 2: IEEE Standard 1540 for software risk management [18]

4. RESEARCH METHODOLOGY

4.1 Questionnaire

A questionnaire with nine questions was used to evaluate the respondents' opinions on aspects of risk management in software projects in South Africa. The nine questions were designed to enable the five research questions mentioned in section 1.2 to be answered. In

total, 56 questionnaires were sent to various software professionals working within the following industries in South-Africa:

- Defence
- Mining
- Banking
- Enterprise software

A total of 35 questionnaires were returned; two of these were completed by individuals not actively working on software projects, and were discarded. The result is an effective recovery rate of 59 per cent (33 out of 56). Figure 3 below shows the distribution of the positions held by the 33 respondents.

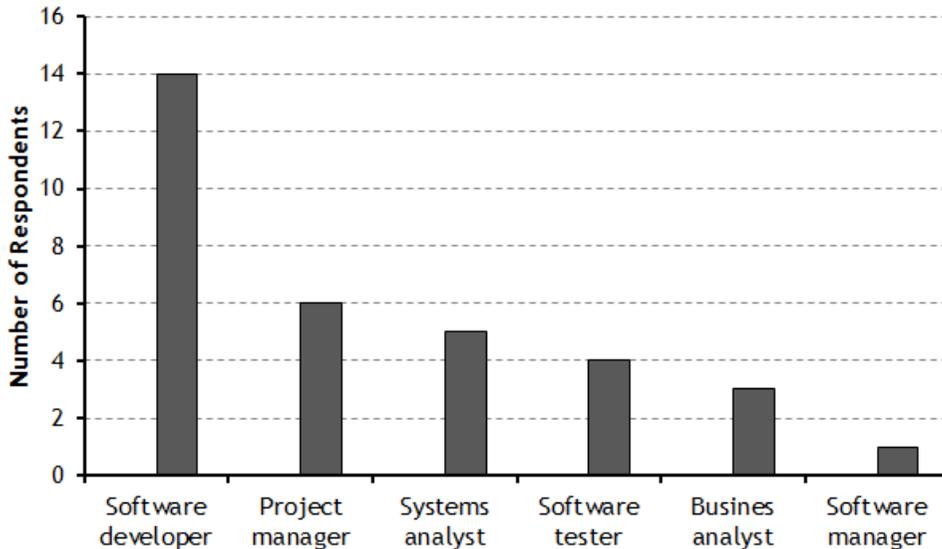


Figure 3: Positions held by respondents to questionnaire

It can be seen from Figure 3 that nearly half of the respondents were software developers.

4.2 Hypotheses

The following hypotheses were postulated for this study.

4.2.1 Nature of risks

- H0: Software project risks faced by organisations in South Africa are not the same as those in the developed world
- H1: Software project risks faced by organisations in South Africa are the same as those in the developed world

4.2.2 Impact of risk management on project success

- H0: Risk management has no impact on project success
- H1: Risk management increases the probability of project success

4.2.3 Formal risk management procedures vs ad hoc risk management procedures

- H0: Formal risk management procedures do not produce better results than ad hoc risk management procedures
- H1: Formal risk management procedures produce better results than ad hoc risk management procedures

5. RESULTS

The first question in the questionnaire requested information on the position held by the respondent; the distribution was shown in Figure 3. In some cases, more than one question was used to answer one of the five research questions. The following section reports on the results for each research question.

5.1 Research question 1

In order to answer research question 1, respondents were asked to answer questions 2 to 4 of the questionnaire.

Question 2: What percentage of software projects is completed on time, within budget and with all the required functionality?

The results for this question are summarised in the distribution illustrated in Figure 4.

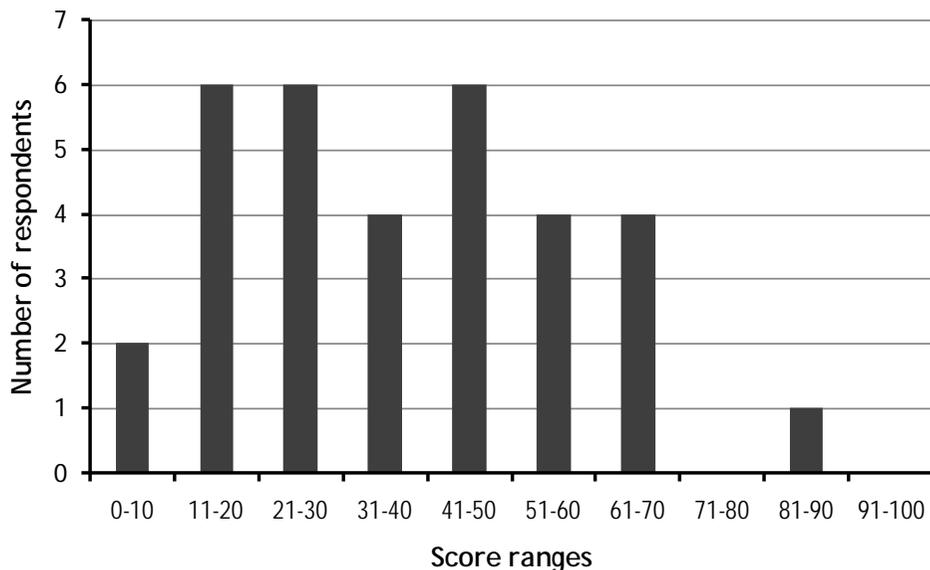


Figure 4: Percentage of software projects that are successful

The mean value for the respondents' scores was 37.1 per cent, and the standard deviation was 20.1 per cent. Only one respondent was of the opinion that more than 70 per cent of software projects were successful.

Question 3: What percentage of software projects fail to meet any or all of the following three requirements, i.e. completed on time, within budget, and with all the required functionality?

The results for this question are summarised in the distribution illustrated in Figure 5.

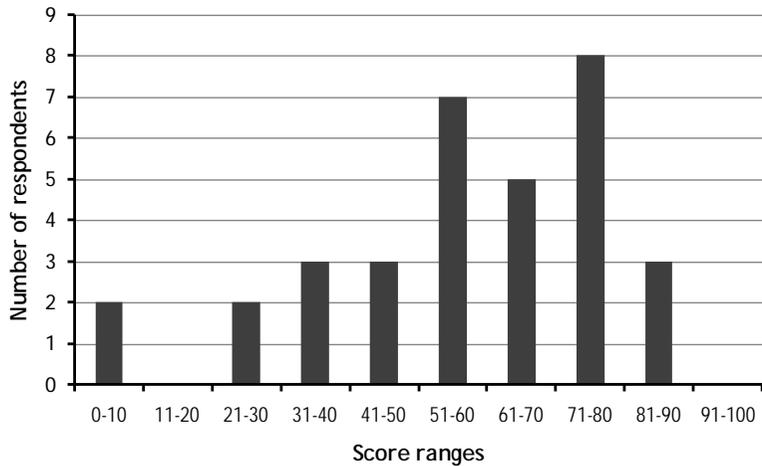


Figure 5: Percentage of software projects that are challenged

The mean value for the respondents' scores was 56.5 per cent, and the standard deviation was 21.4 per cent.

Question 4: What percentage of software projects is terminated before completion?

The results for this question are summarised in the distribution illustrated in Figure 6.

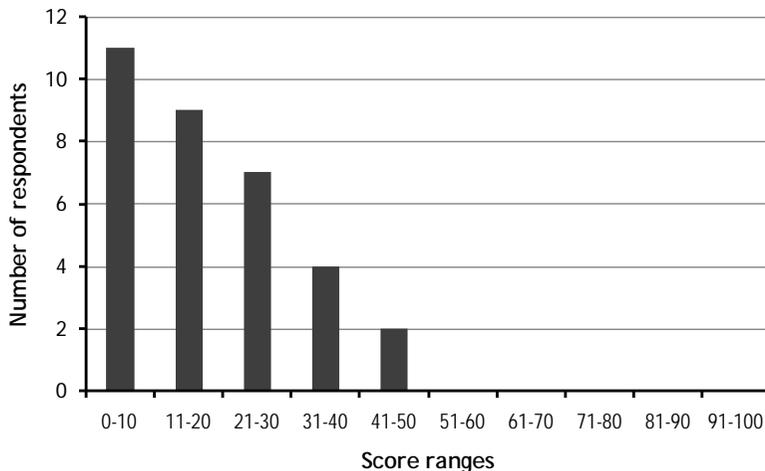


Figure 6: Percentage of software projects that are terminated before completion

The mean value for the respondents' scores was 18.0 per cent, and the standard deviation was 12.4 per cent.

One-sample T-Tests were also applied to the responses from all three questions above, to compare the actual means to the hypothetical means (the average values reported by the Standish Group for successful, challenged, and failed projects). Table 2 below summarises the results of the one-sample T-Tests.

Table 2: One-sample T-Test results

		Actual mean	Hypothetical mean	P Value
Question 2	Successful	37.1 %	32%	0.044
Question 3	Challenged	56.5 %	44%	0.097
Question 4	Failed	18.0 %	24%	0.161

Although the actual mean values for Questions 2 to 4 differ from the Standish Group values, the difference was not statistically significant for any of the three tests. The success rate of software projects in South Africa is therefore roughly the same as in the developed world. There is therefore significant room for improvement in the success rate of software projects in South Africa. The need to investigate improvement measures can thus be substantiated.

5.2 Research question 2

After a thorough exploration of the literature on the definition of risk, it was concluded that a suitable definition must contain the following two properties: 1) uncertainty (an event that may or may not occur) and 2) loss (an event that has undesired results). The following definition of risk by Kontio [8] was found to be clear, concise, and applicable to the South African and the global software industry.

“Risk is a possibility of loss - or any characteristic, object or action that is associated with that possibility”

5.3 Research question 3

In order to answer research question 3, Question 5 of the questionnaire provided a list of Boehm’s top 10 software risk factors [5], and respondents had to answer ‘Yes’, ‘No’, or ‘Not sure’ to whether they have experienced each one of Boehm’s 10 factors as particularly risky within the South African context. The ‘Not sure’ option was added to reduce the number of inaccurate responses. The results are summarised as stacked columns in Figure 7 below.

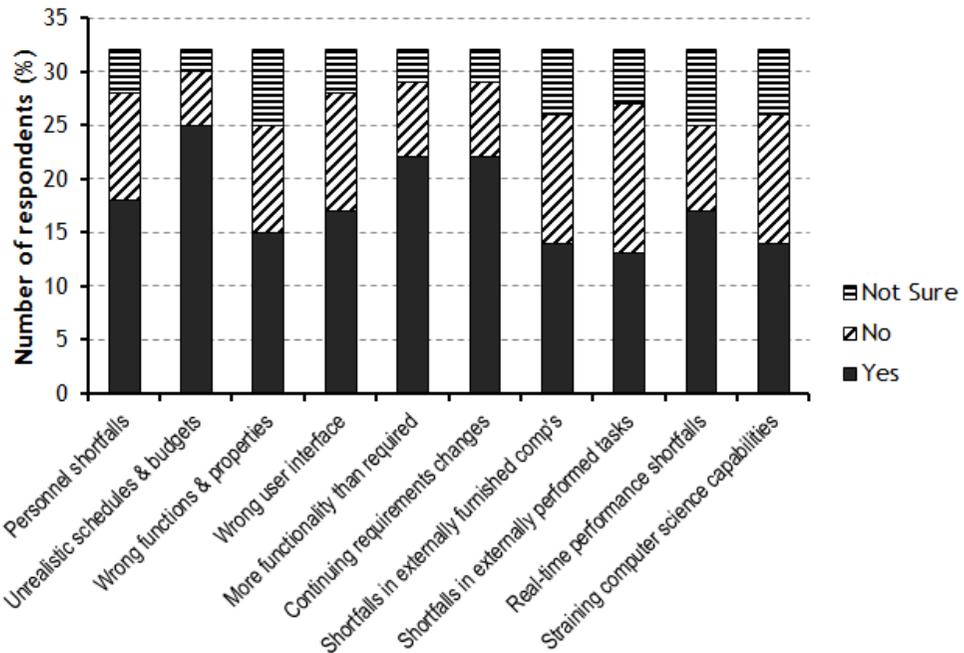


Figure 7: Presence of risk factors for South African software projects

The following three factors are perceived to be the 'riskiest' for South African software companies.

- Unrealistic schedules and budgets
- More functionality than required
- Continuing requirements changes

If the 'Not sure' responses are disregarded, the 'Yes' responses represent 64.8 per cent of the total 'Yes' and 'No' responses. A one-sample T-Test was applied to the responses to test whether the sampled mean of 64.8 per cent is statistically significant, being higher than the hypothetical mean of 50 per cent. (A score of 50 per cent or larger is regularly considered worldwide to be the target figure for passing tests.) Table 3 below summarises the results of the T-Test.

Table 3: T-Test results

Actual Mean	Target Mean	P value
64.8 %	50 %	0.0036

The results of the T-test produced a P-value of 0.0036, which is statistically significant as it is greater than the target of 50 per cent. The null hypothesis (H0) can thus be rejected and the research hypothesis (H1) accepted. The results therefore indicate that software development projects in South Africa face the same risks as similar software projects in the developed world.

5.4 Research question 4

In order to answer research question 4, respondents were asked to answer Questions 6 and 7 of the questionnaire.

Question 6: What impact does risk management have on the success of software projects in terms of completing the project on time, within budget and within scope?

Respondents had to select one of the following three options:

- Risk management increases the chance of success
- Risk management has no impact on project success
- Not sure

A chi-squared test was applied to the results. A target of larger than 50 per cent was set for responses that declared a positive impact as a consequence of risk management. The 'not sure' responses were once again discarded. The results are shown in Figure 8 below.

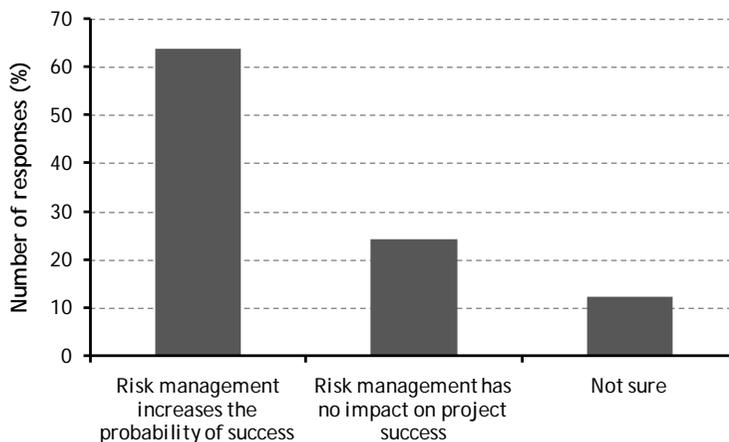


Figure 8: Impact of risk management

The results of the chi squared test produced a P-value of 0.0258, which is statistically significant. The null hypothesis (H0) can thus be rejected and the research hypothesis (H1) accepted.

It can therefore be stated that '*Risk management increases the probability of project success*' for software projects.

Question 7: Did the applied risk management procedures contain any of the following steps (as identified by the IEEE standard for software risk management)?

- a) Risk management planning (Planning how the risk management process will be approached and implemented)
- b) Managing the risk management profile (Continuously monitoring and documenting the state of all risks as well as keeping the stakeholders informed on the state of risks)
- c) Performing risk analysis (Risk identification, estimation, evaluation)
- d) Performing risk treatment (Selecting risk treatment, planning and implementing risk treatment)
- e) Risk monitoring (Continuously monitoring risks and risk treatments as well as seeking new risks)

Respondents had to choose from three options, i.e. 'Yes', 'No' or 'Not sure'.

The results of this question produced a mean of 40 per cent of 'Yes' responses. The intention was to test whether the sampled mean was statistically significant in being larger than the target mean of 50 per cent. It is clear that the sampled mean was not larger than the target mean. No further statistical analysis was thus required. This means that risk management procedures in South Africa typically do not conform to the IEEE standard for software risk management, and cannot be considered as adequate.

5.5 Research question 5

In order to answer research question 5, participants were asked to answer Questions 8 to 10 of the questionnaire.

Question 8: 1. Do formal risk management procedures typically yield better results (with regard to completing a project on time within scope and within budget) than ad hoc internally developed procedures?

Respondents had to choose from the following three options:

- Yes - Formal methods produce better results than internal methods
- No - Formal methods do not necessarily produce better results than internal methods
- Not sure - I'm not in the position to accurately compare the two methods.

The responses to this question are illustrated in the graph in Figure 9 below.

A chi squared test was applied to the responses. A target of larger than 50 per cent was set for responses that declared formal procedures to perform better than internal procedures. The 'Not sure' responses were discarded.

The chi squared test produced a P value of 0.6459, which is not statistically significant. The null hypothesis (H0) could not be rejected. Therefore it cannot be stated that formal risk management procedures produce better results than internal methods.

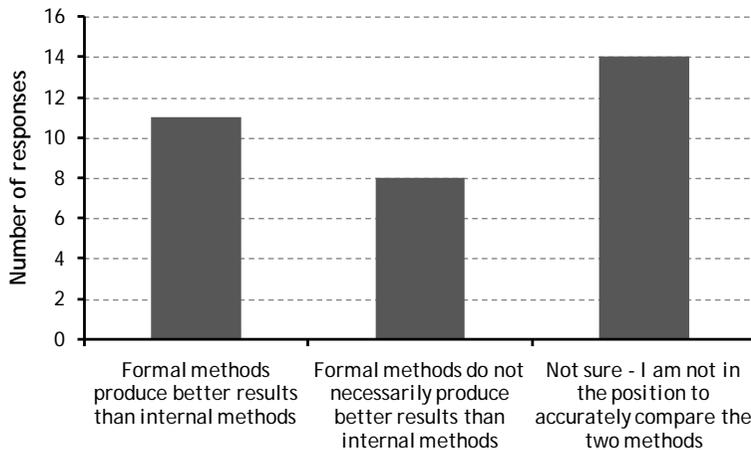


Figure 9: Formal vs ad hoc risk management

Question 9: If formal risk management procedures were used on some of your projects, name them (if none were used, leave blank).

No procedures were mentioned by more than one respondent. No statistical analysis could therefore be performed on the responses.

Question 10: Name the risk management procedure that produced the best results with regard to completing a project on time, within budget, and within scope (if no one procedure produced notably better results than others, leave blank).

No procedures were mentioned by more than one respondent. Therefore it cannot be stated that one specific formal risk management procedure performs better than others.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The following conclusions can be made with respect to the research questions.

- The success rate (completing the project on time, within budget, and within scope) of software projects in South Africa is low, as it is in the rest of the world. The need to investigate the impact of risk management on project success rates can thus be substantiated.
- The application of risk management processes or procedures increases the chances of successfully executing software projects in South Africa.
- The risks that face software projects are generally the same in South Africa as in the developed world.
- Most South African software organisations use ad hoc internally developed risk management procedures. These procedures mostly do not conform to the IEEE standard [18] for software project risk management. The adequacy of these procedures is thus questionable. Having stated that, the success rate of South African software projects was higher where ad hoc risk management was applied, compared with projects where no risk management was applied.
- The fact that the majority of software risk management methods that were applied to South African software projects were ad hoc internally developed methods, rather than recognised formal methods, means that the effectiveness of formal risk management procedures could not be accurately measured. The results of formal methods could thus not be compared with the results of ad hoc methods.

- No formal procedure was applied to a sufficient number of projects (within the test sample) to measure its results properly. It was thus not possible to state that one procedure is more effective than others.

This study investigated the application of the IEEE Standard 1540 for risk management within software development projects in South Africa. This standard is fairly generic, and the steps are similar to many other risk management processes in use today. The results indicate that formal processes are not always used by organisations. The information engineering (IE) approach was developed to assist organisations to execute software development projects successfully, and to do this on time and within budget. Application of a risk management process should form part of IE; and the results of this study indicate that there is a definite need for a more formal approach to delivering software projects in South Africa more successfully. The IE methodology is therefore not applied in most software projects in South Africa, and probably contributes to the low success rate.

6.2 Recommendations

It is recommended that South African software organisations implement some form of risk management in all of their software projects, since the application of risk management has a positive impact on project success.

Given that software project risks were found to be the same in South Africa as in the developed world, local software organisations are very likely to encounter a number of the risks that were identified as being especially prevalent in the developed world (such as Boehm's top 10 software project risks [5]). It is therefore recommended that South African software organisations take the necessary steps appropriately to mitigate these risks that are common to most software projects.

6.3 Future research

There is still a gap in research into the effectiveness of formal software risk management procedures in South Africa. The fact that risks are the same in South Africa as in the developed world suggests that mitigation strategies that have proven to be successful in the developed world will also be successful in South Africa. This statement, however, cannot be empirically proven at this point. So there is still a need to find organisations that implement formal software risk management procedures, and to measure the effectiveness of these procedures.

REFERENCES

- [1] The Standish Group. 2009. *Standish chaos report*. www.projectsmaart.co.uk/docs/chaos-report.pdf [Accessed on November 11, 2010].
- [2] Fairley, R. 1994. Risk management for software projects, *IEEE Software* 11 (3) pp. 59-67.
- [3] Hogan, P.T & Raja, M.K. 1997. Information engineering implementation in organizations: A study of factors affecting success. *Journal of Information Technology Management* 8 (3,4).
- [4] Misra, S.C., Kumar, V. & Kumar, U. 2006. *Different techniques for risk management in software engineering: A review*. Eric Sprott School of Business, Carleton University, ASAC, Banff, Alberta.
- [5] Boehm, B.W. 1991. Software risk management: Principles and practices, *IEEE Software* 8 (1), pp. 32-41.
- [6] Esteves, J., Pastor, J., Rodriguez, N. & Roy, R. 2004. Implementing and improving the SEI risk management method in a university software project. *IEEE Latin American Transactions*, 3, (1), pp. 90-97.
- [7] Kontio, J. 1997. *The Riskit method for software risk management*, version 1.00, UMIACS-TR-97-38, Institute for Advanced Computer Studies and Department of Computer Science Technical Report, University of Maryland.
- [8] Kontio J. 2001. Software engineering risk management: A method, improvement framework and empirical evaluation. Doctoral dissertation, Helsinki University of Technology, Center of Excellence, ISBN 952-5136-22-1.
- [9] Keshlaf, A.A. & Hashim, K. 2000. A model and prototype tool to manage software risks, *IEEE Quality Software*, 1,(1), pp. 297-305.

- [10] **Software Engineering Institute.** 2007. *Capability maturity model integration, Version 1.2*, Carnegie Mellon University, United States of America. (<http://www.sei.cmu/cmmi>)
- [11] **Williams, R.** 2006. The CMMI RSKM process area as a risk management standard. *Proc. of Sixteenth Annual International Symposium of the International Council on Systems Engineering*. INCOSE, Orlando, Florida, USA.
- [12] **Hall, C. & Fourie, L.** 2007. Exploring the role of the human resource function in the South African information technology industry. Department of Human Resource management, University of Johannesburg. *SA Journal of Human Resource Management*, 5, pp. 54-64.
- [13] **Goguen, A., Stoneburner, G. & Feringa, A.** 2002. *A risk management guide for information technology systems*. National Institute of Standards and Technology, <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf> [March 28, 2011].
- [14] **Thomsett, R.** 2004. *Risk in projects: Total tool set*, The Thomsett Company Publications.
- [15] **Roy, G.G.** 2004. *A risk management framework for software engineering practice*. School of Engineering Science, Murdoch University, Perth, Australia.
- [16] **Samad, J. & Ikram, N.** 2006. *Managing the risks: An evaluation of risk management processes*. Javeria Centre for Software and Systems Engineering. Muhammad Ali Jinnah University.
- [17] **De Wet, B.** 2011. An evaluation of software project risk management in South Africa, Project Report for the Degree M. Sc. (Technology Management), Department of Engineering and Technology Management, University of Pretoria.
- [18] **IEEE Standards Association, IEEE 1540-2001, 2001.** *IEEE standard for software life cycle processes - Risk management*. The Institute of Electrical and Electronics Engineers, Inc. ISBN 0-7381-2834-1.