Investigating the Application of Risk Management in Greek IT Companies

Tsiara Eleana[1], Kerstin Siakas[2]

[1]Technological Educational Institute of Thessaloniki
Department of Informatics
P.O. Box 141, 57400 Thessaloniki, Greece
etsiara@it.teithe.gr

[2]Technological Educational Institute of Thessaloniki
Department of Informatics
P.O. Box 141, 57400 Thessaloniki, Greece
siaka@it.teithe.gr

Abstract

The persistent software crisis has led to many different approaches for developing software projects on time, within budget and according to customer requirements.

The objective of risk management is to identify potential problems before they occur and to plan and carry out risk-handling activities in order to mitigate potential risks. Risk management is a continuous, forward-looking process that must be an integral part of day-to-day business activities and processes. Risk management should address issues that could endanger achievement of critical objectives. A continuous risk management approach is applied to effectively anticipate and mitigate the risks that have critical impact on the project.

In this paper, we investigate the level of risk management application in 86 Greek IT companies. For this purpose a structured questionnaire was designed. This paper aims to present the fundamental elements of the risk management concept and reveal the results of the analysis.

Keywords: Risk, Risk Management, Risk in IT Projects, Risk in Software Projects, Risk in IT Companies, Risk in Software Companies

1.0 Introduction

The science of informatics has been developing rapidly since the late 1970s. Today, software owns a significant percentage in the overall economic activity and could be considered an industrial product [1].

Software projects have some characteristics that makes their development more difficult, in comparison with other products. First of all, software is intangible, so the developing progress and its behaviour is not directly visible. Second, it has high complexity and third, it is flexible and is constantly subject to changes, even after the developing process is over, as it is necessary to comply with the changes in its environment [1, 2].

According to recent researches, a percentage of 31.1% of software projects fails completely, while an even bigger percentage (52.7%) is completed, but outreaches the time and cost limits, diverging from the requirement that have been defined, providing less, than the expected, functionality and its evolution, in order to satisfy the new needs that come up, becomes impossible [1,3].

To increase the probability of successful completion for software projects, several methods and techniques have been developed, that emphasise on the management of the software development process. Managing a software project includes activities like project planning, which includes defining the processes that require implementation, time scheduling and cost evaluation, providing guidance for the development process with instructions, monitoring the project progress, and keeping it under control, by taking appropriate corrective actions [4].

Risk management is an important managing activity, as one of the major factors that defines a project's success are risks and the degree that the company is making efforts to handle them [5]. Many software projects could have been successfully completed and have a full functionality, according to the defined requirements, if the risks had been taken into account during the developing process, and if the company had tried to eliminate them.

The basic risk in a software project is that the latter is not completed in appropriate time, overruns the expected cost [2] and is not being developed according to the plan, without, of course, excluding all the other risks that any company is exposed to.

2.0 Risk Definition

Below, there are two definitions for risk provided by [6]:

- «*A possible future event that, if it occurs, will lead to an undesirable outcome*»
- «*Risk refers to a possibility of loss, the loss itself, or any characteristic, object, or action that is associated with that possibility*»

The term "*risk*" may be partially different, depending on the section it is used for. In IT companies it is an important issue, often referring to the sources of risk for owning, developing and maintaining software. The meaning of risk is directly connected to uncertainty. Uncertainty is present at every business venture that regards product development, as there are unknown factors that can lead to unexpected results. These factors can be favourable or not, accompanied by a possibility of profit or loss, respectively [7].

3.0 Project Risks Categories

Risks can be divided into the following categories [8]:

Technological risks. Risks can be caused by late delivery or unavailability of the necessary mechanical equipment, the use of inappropriate tools, the use of recent and probably not sufficiently tested technology, the unavailability of computer resources and the insufficient facilities [4].

Technical risks. They are related to software performance and involve problems with the programming languages that are being used, the project size and functionality, the platforms which the software is compatible with, the quality and reliability, and also the time required to complete each development process.

Process risks. They rise from excessive restrictions, lack of experience or skills of the development and management team, insufficient staff education and non-effective communication and collaboration.

Financial risks. They include cost prediction, along with the profit and loss limits. They are related to software cost, from the beginning of the development until the delivery.

Human risks. They have to do with the human resources of the company and include matters like the lack of experience, conflicts between the employees, ethical issues and low productivity.

Time scheduling and scope risks. Changes that occur in time scheduling and scope after the development has begun, are very common in software projects, so at some point, they may be considered rational.

4.0 Risk Management Origin

Risk management comes from the probability theory and the decision making under conditions of uncertainty. Specifically, it was significantly influenced by the expected utility theory, the theory of bounded rationality and prospect theory. Expected utility theory refers to the fact that people make a choice among different alternatives based on the expected utility of each one. Theory of bounded rationality states that in the real world, different results and their probability cannot be easily realized by people. Prospect theory helps modelling the influence of

human perception on making a decision [6]. The study of risk management began after World War II and in the beginning it aimed at protecting people and companies from losses that came out of accidents. During the 1950s, new types of risk management made their appearance, while in the 1960s there were developed activities focusing on handling unexpected events. The following decades of 1970 and 1980 second-generation types of risk management were implemented and economic management was enhanced. It was after 1990, when functional risk management and liquidity risk management emerged. Furthermore, during the same period addressing risks started spreading throughout the globe [9].

5.0 Risk Management Definition

According to NIST (National Institute of Standards and Technology), risk management is the procedure of identifying risks, evaluating them and following steps to reduce these risks to a tolerable level [10]. In IT companies in particular, risks must be addressed before they turn into a threat for software's proper operation or a point is reached where extremely expensive rectifications are required. So the goal of this management is reducing risks and their impact, without eliminating them completely [7].

The process of risks management includes assessing risks and taking appropriate measures that aim at resolving them [11]. First of all, each project activity is observed and the risks that decrease the project's chances of success are discovered. Every risk is evaluated based on its attributes, such as probability of occurrence, negative impacts and severity. Afterwards, the actions that need to be done so as to avoid or mitigate the identified risks when this is necessary are defined [2].

The procedure is repeated during the whole life cycle of the software project, as new information is revealed and new risks appear [4]. The preconditions for the efficient function of risk management is that all members of the development and management group take its specifications into account and are capable of identifying the risks as soon as possible in the project's life cycle and take care for their immediate resolution [2, 12]. This is achieved with the continuous and effective communication between the members and by providing a channel for information exchange between them and the stakeholders, as communication constitutes the cornerstone of managing risks successfully [11, 13].

6.0 Risk Measures

Software measures are necessary in order to evaluate risks, thus it can be defined if the project is close to approaching its goal. Risk measures offer both subjective and objective data, which can be used to indicate the risk level of the whole project. Below, there are explained some measures, that will be mentioned later [14]:

Risk category. The numbers of risks that have been identified in each category, indicates how much specific types of risks may affect the project.

Risk exposure. It is defined as RE = P x C, where RE is risk exposure, P is the probability of an undesirable outcome and C stands for the consequences of this outcome.

Risk leverage. It is defined as (REbefore – REafter) / [risk resolution cost], where REbefore is the RE before starting the resolution process and REafter is the RE after the process is completed. Consequently, risk leverage is a measure of the relative cost, when several activities for risk resolution are being implemented.

Risk threshold. It is specified relying on a quantitative objective, for every risk parameter. A risk is acceptable as long as its value does not overcome the threshold.

Annualized loss expectancy. It is the expected monetary loss due to a risk over a one year period. It is calculated by summing the loss for every asset of the company, because of a single risk, multiplied by the risk's occurrence rate.

Return on investment. The economizing achieved by managing one or more risks, divided by the management cost [15].

7.0 Risk Management Approaches

Just like any other strategy, risk management employs particular approaches for the purpose of accomplishing its goals. There have been developed different models, each one with specific procedures, which focus on different issues.

Table 1: Comparison of six basic risk management models

Model	Characteristics	Complexity	Focus
Barn-Boehm Theory	Risk identification and maintaining a top 10 risk list	Low	Risk identification
Riskit Method	Focuses on the project's goal and investors, Riskit analysis graph	High	Analysis of risks and their components
SoftRisk Model	Constant identification and risk control, it maintains a list with the top 10 risks and risk statistics	Middle	Risk statistics
IEEE (Institute of Electrical and Electronics Engineers) Risk Management Standard	Continuous risk management, continuous improvement, standards development, being able to manage project and organizational risks	Middle	Risk management description table
CMMI (Capability Maturity Model Integration) Risk Management Process Area	It includes specific stages, continuous improvement of the model, staff training	Middle	Risk database
Continuous Risk Management (CRM) Model of Software Engineering Institute (SEI)	Continuous risk management, emphasizes on risk communication	Low	Risk communication

The approaches mentioned in table 1 are based on [16, 17, 18, 19, 20]. Continuous Risk Management (CRM) is a risk management practice with specific processes, methods and tools [21].

8.0 Research Implementation and Analysis

8.1 Previous Research

Generally, there has been little research on gathering statistical data about risk management concerning software projects. Two papers that seemed to be abundant on this matter and were used as guidance for the current study are [22, 23]. No previous research has been discovered in the context of Greek software companies.

8.2 Research Objectives

The objective of the study was to estimate the percentage of usage and the contribution of risk management in the successful completion of software projects. Specifically, the purpose was to provide insight on the following topics:

- How often risk management is used in software projects;

- How many projects fail when using and when not implementing risk management;
- How effective are formal and ad hoc risk management approaches.

8.3 Demographic Data

The study was anonymous, was conducted between March and July 2015 and involved Greek companies or international companies with a Greek branch that engage in developing software. The majority of the companies were mainly involved in web designing. Overall, 85 responses were gathered. The person who responded was asked to be a project manager or somebody with appropriate knowledge for completing the questionnaire. Moreover, we requested that only one response was returned per company.

The majority (75) of the sample consisted of small companies with less than 10 employees. Five companies had 10-49 employees and the remaining five 50-250 employees.

37 of the 85 companies were relative new, being active less than 5 years, while 32 companies were active 5-10 years and 16 were founded more than 10 years ago.

8.4 Risk Management Usage and Failure Rate

In total 47 out of the 85 companies claimed to be using a strategy for managing risks.

The prominence of ad hoc approaches over formal techniques was evident, as 31 companies used informal methods and 16 preferred formal techniques. Specifically, CRM model of SEI was used by 7 respondents, IEEE Standard by 5, CMMI model by 2, and Riskit and SoftRisk model by one company each.

Figure 1 shows that:

- 34 respondents of the total 38 companies, who do not use risk management, claim that the failure rate is less than 25%, one claims it is over or equal to 25 and below 50%, two claim it is 50%-75% and one greater than 75%. The average failure rate was estimated to be 17.76%
- 15 companies out of the 16 that followed a formal risk management technique as a part of the software development process, stated that the failure rate is less than 25% and one claimed it is over or equal to 25 and below 50%, while no one claimed that is reaches or exceeds 50%. The average failure rate was calculated 14.06%
- concerning the 31 users of an ad hoc method, 24 reported a failure rate below 25%, 4 stated that is was over or equal to 25 and below 50%, 2 that it was between 50% and 75%, and finally one admitted that the rate exceed 75%. As it emerges the average failure rate is 21.37%

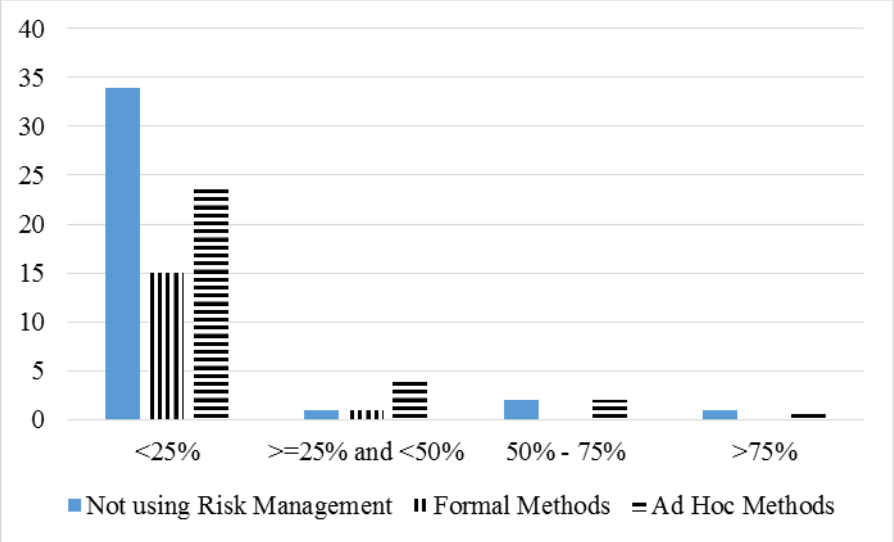

Figure 1: Percentage of failures/cancellations without the usage of risk management, with the usage of a formal risk management method and the usage of an ad hoc method

The results show that there is a slight difference among companies that did not use any risk management approach, companies that opted for a formal method for managing risks and respondents that preferred an ad hoc, tailor-made approach.
For the purpose of validating this conclusion mathematically, responses for the three groups of users were divided in two categories: the first containing those who reported a failure rate below 25% and the second merging all answers that were equal or above 25%, as Table 2 depicts.

Table 2: Categorization of responses for a failure rate below and above or equal 25%

	Not using Risk Management	**Formal Methods**	**Ad Hoc Methods**
<25%	34	15	24
>=25%	4	1	7

Following, a two-tailed Fisher's exact test was implemented for all the pairs of users. The P-values produced are presented below:

- Not using risk management - Users of formal methods: 1
- Not using risk management - Users of ad hoc methods: 0.2
- Users of formal methods - Users of ad hoc methods: 0.23

Since all P-values were exceeding 0.05 it can be deduced that none of the associations is statistically significant.

Most of risk management users certify its effectiveness concerning the successful completion of software projects (Figure 2). In detail, 40 respondents perceived that it increases the possibility of a successful outcome, 1 stated that is has no effect and 6 were not sure regarding the impact of risk management.

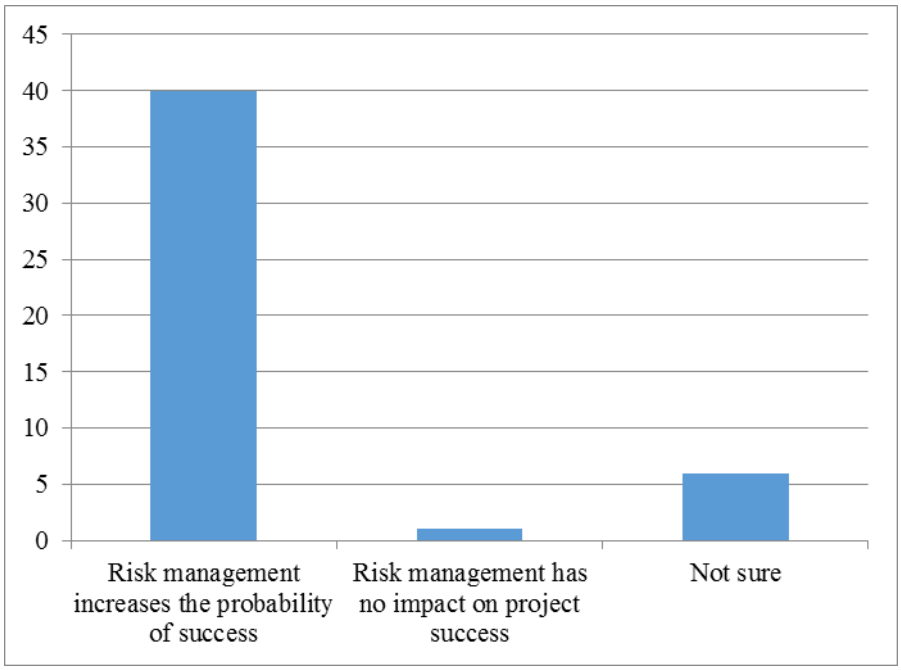

Figure 2: Risk management impact on software projects success for users of the risk management approach

8.5 Research Restrictions

It is presumable that the responses were not completely accurate, despite the fact that we guaranteed that all the information submitted would remain strictly confidential and it was not possible for us to identify any of the companies, which participated in the research. This is a general challenge when using questionnaires regarding confidential data.

Specifically, we are very sceptical concerning the questions related to the percentage of the unsuccessful projects, as the companies may have concealed the true answers, by reducing the number of failed projects, in order to avoid revealing sensitive data that would hypothetically hurt their reputation.

9.0 Research Conclusions and Discussion

The percentages of failed software projects among companies which did not follow a risk management strategy, companies that used a specific model and companies that managed risks with a custom-made approach were hardly diverse.

Thus, from this survey it can be concluded that using risk management did not provide a notable advance in terms of successful software completion and no significant difference was observed in failure rates between users of a formal and ad hoc risk management technique.

10.0 References

1 Giakoumakis Manolis, Diamantidis Nikos (2009). Software Technology (Τεχνολογία Λογισμικού), Stamouli A.E. (Σταμούλη Α.Ε.).

2 Hughes Bob, Cotterell Mike (1999). Software Project Management (Second Edition), McGraw – Hill.

3 The Standish Group (2014). Chaos, Project Smart.

4 Sommerville Ian (2009). Basic Principles of Software Engineering (Βασικές Αρχές Τεχνολογίας Λογισμικού), Κλειδάριθμος.

5 Dorsey Dr. Paul (2005). Top 10 Reasons Why Systems Projects Fail, Dulcian Inc.

6 Misra C. Subhas, Kumar Vinod, Kumar Uma (2006). Different Techniques for Risk Management in Software Engineering: A Review, ASAC, Banff, Alberta.

7 Ravindranath C. Pandian (2007). Applied Software Risk Management: A Guide for Software Project Managers, Taylor & Francis Group, Aurbach Publications.

8 Avdoshin M. Sergey, Pesotskaya Y. Elena (2013). Software Risk Management: Using the Automated Tools.

9 Georges Dionne (2013). Risk Management: History, Definition and Critique, CIRRELT-2013-17.

10 Netland Lars-Helge (2008). Assessing and Mitigating Risks in Computer Systems.

11 Williams Laurie (2004). Risk Management. Retrieved 28th January 2016, from: http://agile.csc.ncsu.edu/SEMaterials/RiskManagement.pdf.

12 Morisio Maurizio, Egorova Evgenia, Torchiano Marco (2007). Why software projects fail? Empirical evidence and relevant metrics, IWSM – Mensura.

13 Dhlamini John, Nhamu Isaac, Kachepa Admire (2009). Intelligent Risk Management Tools for Software Development.

14 Hall M. Elaine (1998). Managing Risk: Methods for Software System Development, Software Engineering Institute, Addison Wesley Longman Inc.

15 Georgiou Sophia (2012). Analysis and Risk Management in Information Systems – Methodology Implementation in Business Environment (Ανάλυση

και Διαχείριση Επικινδυνότητας στα Πληροφοριακά Συστήματα – Υλοποίηση Μεθοδολογίας σε Επιχειρησιακό Περιβάλλον), PhD Thesis.

16 Wan Jiangping, Wan Dan, Zhang Hui (2010). Case Study on Business Risk Management for Software Outsourcing Service Provider with ISM, Technology and Investment, Vol. 1, No. 4, 10 pages.

17 Papadimitriou Athanasios (2009). Risk Management in IT Projects (Διαχείριση Κινδύνων σε Έργα Πληροφορικής), PhD Thesis.

18 Williams Ray C. (2006). The CMMI RSKM Process Area as a Risk Management Standard, Sixteenth Annual International Symposium of the International Council on Systems Engineering (INCOSE).

19 Pan Chun-guang, Chen Ying-wu (2008). An Optimization Model of CMMI-Based Software Project Risk Response Planning, World Academy of Science, Engineering and Technology, Vol. 2, No. 2, pp. 637-641.

20 Stern Robert, Arias José Carlos (2011). Review of Risk Management Methods, Business Intelligence Journal, Vol. 4, No. 1, pp. 59-78.

21 Dorofee J. Audrey, Walker A. Julie, Alberts J. Christopher, Higuera P. Ronald, Murphy L. Richard, Williams C. Ray (1997). Continuous Risk Management Guidebook, Software Engineering Institute, Carnegie Mellon University.

22 De Wet B., Visser J.K. (2013). An Evaluation of Software Project Risk Management in South Africa, South African Journal of Industrial Engineering, Vol. 24 (1), pp. 14-28.

23 Bannerman L. Paul (2008). Risk and Risk Management in software projects: A reassessment, The Journal of Systems and Software 81, NICTA, Australian Technology Park, Eveleigh, NSW, Australia, pp. 2118–2133.