Risk Management in Software Development: A Technology Overview and the Riskit Method

Jyrki Kontio
http://wwwseg.cs.hut.fi/

Nokia Telecommunications
IP Networking
P.O. Box 315
00045 NOKIA GROUP, Finland
Tel: +358-9-5116-3233
jyrki.kontio@ntc.nokia.com

Helsinki University of Technology
Department of Computer Science and Engineering
P.O. Box 1100
FIN-02015 HUT, Finland
Tel: +358-9-451-4852
jyrki.kontio@cs.hut.fi

ABSTRACT
Explicit and systematic management of risks in software projects has become a more common practice amongst leading software organizations. However, often the methods used have severe theoretical and practical limitations that may lead to biased or inappropriate control of risks. The first part of this tutorial presents a critical overview of the current risk management technology, discussing the pros and cons of main approaches, as well as guidelines for their use. The second part of the tutorial presents the Riskit method with concrete examples and exercises. Riskit is a risk management method that has been developed to provide a theoretically sound, yet practical risk management approach. The method has been used and evaluated in several industrial projects in Europe and in the U.S.

Keywords
Risk management, project management

1 INTRODUCTION

All software development projects involve risks and, in fact, ability to take and manage risks is a critical success factor in managing software related businesses. It seems that majority of software managers manage risks intuitively. However, as intuitive risk management is perceived as unreliable and inconsistent way of dealing with risks, more systematic risk management programs and methods are gaining ground in the industry, as evidenced by the many methods, tools and reports that are available (e.g. [5]).

ICSE '99 Los Angeles CA

While many of the current risk management approaches work quite well in practice, many of them contain serious sources of bias and, in some cases, are based on flawed or incorrect assumptions. Although many of these shortcomings may not be serious in all situations, practitioners should be aware of them so that the risk analysis results are not compromised.

It seems that the software risk management community has not fully taken advantage of the relevant work on risk management in many other disciplines.

According to our experiences, some of the most common problems or potential biases include the following:

- Communication is hindered by lacking clearly defined and accurate terms. Ambiguous or conflicting terms are used to refer to conceptually different aspects of risks.
- The risk prioritization is often biased due to problems in quantification of risks. E.g., table-based ranking techniques often multiply ordinal scale data to obtain risk rankings, an operation that is not mathematically valid.
- The problems of non-linear utility function and its impact on risk prioritization are rarely addressed by risk management methods.
- Different stakeholder perspectives are seldom explicitly addressed in the risk management process.

In short, the set of tools used for risk management is a very mixed bag of tricks and there seems to be little discussion and attention spent on discussing these potential pitfalls.

2 THE RISKIT METHOD

Riskit is a comprehensive risk management method that is based on sound theoretical principles and thus it avoids many of the limitations and problems that are common to many other risk management approaches in software engineering. As the Riskit method has been extensively presented in other publications [6-8], we present here only the highlights and main principles of the method.

Complete Process Definition
The Riskit method has a comprehensive process definition that supports risk management activities throughout the project [6]. The Riskit process is similar to many other risk management process descriptions with some special characteristics, such as full operational definition of the process and specific steps for defining risk management mandate as well as goals and stakeholders.

Goals and Stakeholders
Most risk management methods do not explicitly support different stakeholder perspectives. Boehm's Win-Win approach is the only major risk management approach that focuses on stakeholder goals [3]. The Riskit method extends Boehm's approach by maintaining links between risks and stakeholders explicitly.

Definition of Risk
The Riskit method supports unambiguous definition for risks. The *Riskit analysis graph* is a graphical formalism that is used to define the different aspects of risk more formally. The Riskit analysis graph can be seen both as a conceptual template for defining risks, and a well-defined graphical modeling formalism. The underlying conceptual model -- or meta-model -- of the Riskit Analysis Graph components is presented in Figure 1.

Practical Application of Utility Theory
The importance of utility theory in decision making is well established in other disciplines [1], and while the concept has also been presented in software engineering risk management [2,4], to our knowledge, it has not been made operational in any major software risk management approach. Ignoring the impact of *utility loss* may seriously influence risk prioritization results. In most situations people and organizations have *non-linear* utility functions w.r.t. observable metric or attribute in question. In other words, the true benefit felt by a stakeholder does not have a linear function to, e.g., money, schedule or defect rate. The Riskit method has incorporated the utility theory components into a straight-forward approach that can be used by practitioners without deeper knowledge of the utility theory.

SUMMARY
The Riskit method combines sound principles into a consistent process and set of techniques. However, the use of these techniques is, of course, not limited to the use of the full Riskit method. Many of the principles and techniques in Riskit can be used to enhance or improve current risk management practices in an organization.

The Riskit method has been evaluated in a number empirical studies in several organizations during the past few years. While the findings from these studies are not conclusive, the empirical feedback indicates that the method is feasible in practice and it seems to result in more detailed analysis and description of risks and it seems to increase the confidence in the results of risk management process.

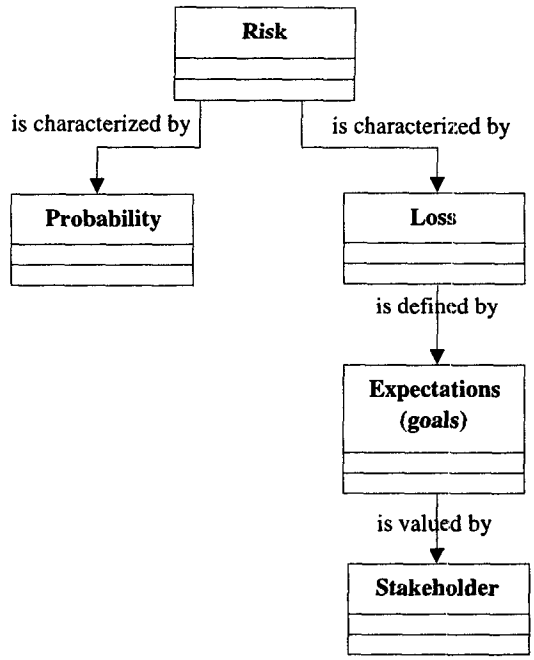

Figure 1: Definition of risk in the Riskit method

REFERENCES

1. P.L. Bernstein. *Against the Gods*, New York: John Wiley & Sons, 1996.
2. B.W. Boehm. *Software Engineering Economics*, Englewood Cliffs, N.J.: Prentice Hall, 1981.
3. B.W. Boehm and Bose P., *A Collaborative Spiral Software Process Model Based on Theory W* 1994. Proceedings of the 3rd International Conference on the Software Process. IEEE Computer Society. Washington, DC.
4. R.N. Charette. *Software Engineering Risk Analysis and Management*, New York: McGraw-Hill, 1989.
5. IEEE, Managing Risk *IEEE Software*, vol. 14, no. 3, 1997.
6. J. Kontio, *The Riskit Method for Software Risk Management, version 1.00* CS-TR-3782 / UMIACS-TR-97-38, 1997. Computer Science Technical Reports. University of Maryland. College Park, MD.
7. J. Kontio and V.R. Basili, *Empirical Evaluation of a Risk Management Method* 1997. Proceedings of the SEI Conference on Risk Management. Software Engineering Institute. Pittsburgh, PA.
8. J. Kontio, G. Getto, and D. Landes, *Experiences in improving risk management processes using the concepts of the Riskit method* pp. 163-174, 1998. Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6).