DRAFT

full paper published in:

proc of. 11th European Software Process Improvement Conference
EuroSPI'2004
November 10-12, 2004 Trondheim, Norway

Paper published in the Industrial Proceedings and presented at the conference

Risk-driven Software Process Improvement - a Case Study

Jakub Miler[1], Janusz Górski
Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdansk, Poland
{jakubm, jango}@eti.pg.gda.pl

Abstract

The paper reports on a case study of practical application of a risk-driven software process improvement framework in a real-life software project. The framework assumes explicit modelling of the process and its deficiencies (risk factors) as well as provides for process evolution. It also includes dedicated techniques to identify process risks and to derive from them suggestions for process improvement. The techniques are embedded in a recurring procedure involving process modelling, risk identification and process improvement steps. The paper presents the case study objectives and reports on the results of two phases aiming at process improvement. Finally, it discusses the case study results as well as presents the issues open for further research.

Keywords

process modelling, risk identification, process improvement, case study

[1] Research partially supported by the grant no. 4 T11 C 008 25 from Polish Research Council (KBN)

1 Introduction to Risk-driven Software Process Improvement

The aim of any software project is to provide the stakeholders with a satisfactory software-based solution of their problem within the schedule and budget limits. The risk of poor product quality and schedule or budget overruns is high which is confirmed by a number of cancelled, delayed or overpaid projects. Effective management of those risks is presently perceived as one of the most important areas of project management [1, 10]. Still, current software processes leave a considerable space for improvement. As process improvement aims at maximizing process quality and effectiveness while minimizing its risks, therefore the support for identification of the most risky process areas and their potential improvement is especially worthwhile.

Current risk identification practices adopt primarily two techniques: checklists and group effort (e.g. brainstorming). Checklists such as [3, 5, 13] help to control the identification scope and protect from overlooking significant risks but they are often too general and do not relate well to actual software processes. Group effort studied e.g. by J. Kontio [4] benefits from synergetic use of human intuition and experience but it exhibits problems with scope focusing and control. Consequently, both existing approaches provide limited output aimed at the process improvement.

The paper proposes a framework for the risk-driven software process improvement. The following features characterize the framework:

- explicit process modelling [6, 9] as well as providing for model evolution [2],
- interpretation of model deficiencies as process risks and areas for potential improvement,
- modelling risks with risk patterns [6],
- supporting risk identification by referring to model metrics and consulting referential models [7],
- deriving suggestions for process improvement from identified deficiencies,
- running the improvement implementation and the modelling as a continuous process.

The recurring procedure of continuous risk-driven process improvement comprises the following steps:

1. initial modelling of the original process,
2. discovering the process risks by applying selected risk identification techniques,
3. (risk-driven) selection of process improvements,
4. systematic transformation of the original process model into the improved process model.

The procedure of continuous risk-driven process improvement is shown in Figure 1.

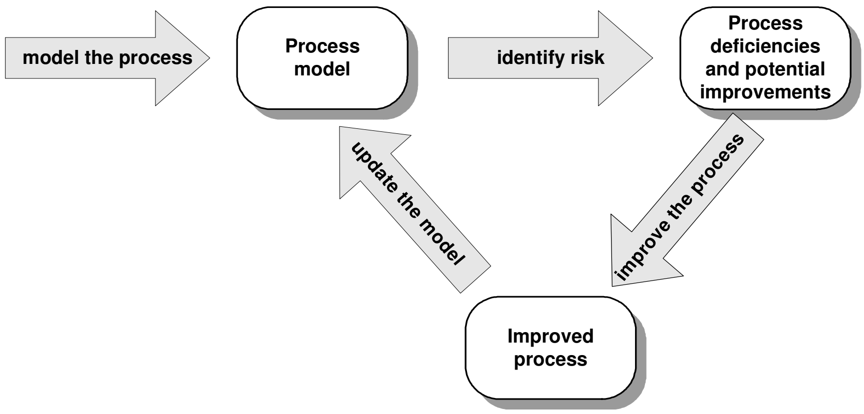

Figure 1: The risk-driven process improvement procedure

2 *Case Study*

2.1 Methodology

The case study aimed at repeated identification of risks in a real-life software project and providing suggestions on possible process improvements.

A software project involving participants from several countries and scheduled for over a dozen months has been chosen for the case study. The project objective was to build a complex, distributed information system based on a novel architecture and business model. During the case study the project remained in the initiation phase. The project description and plans were used in the case study.

The case study comprised 2 phases (cycles of risk-driven process improvement):

- Preliminary risk identification carried out in January 2004 based on formal project description,
- Second risk identification with respect to the improved process carried out in April 2004 based on the Quality Plan, partial Development Plan and the same project description as in Phase 1.

The case study was evaluated by:

- subjecting the risk identification results to the judgement of the project managers based on their intuition and individual experience,
- examining the ratings and priorities the project managers assigned to the identified risks at a risk analysis session,
- assessing the scope of improvement initiated by the project managers after the risk identification.

2.2 Results of Phase 1

To begin with, a process model was built based on the project description. Due to the initial phase of the project, the development process was planned at a rather general level. The most detailed activities covered several months. The project description did not define any qualitative features of the activities, artefacts and roles as those elements were left to be defined later in the plans of the particular project areas. The final model comprised 55 activities at 3 levels of detail, 49 artefacts and 45 roles.

The process risks were first identified using model metrics [7] that were applicable to the model. The metrics indicated two activities and five artefacts for further investigation which resulted in identification of four significant risks. For all those risks, their scenarios have been developed with the help of risk patterns [6]. The example of an identified risk is given below together with the corresponding scenario (first expressed with the help of risk patterns, then expressed as a natural language statement).

Risk factor: *Distributed, multinational development of business models*

Risk scenario (in terms of risk patterns): If **New Business Modelling***<activity>* loses **Consider regional differences in reality***<practice>* then **System Requirements Specification (Vision)***<artefact>* loses **Conformity to target reality***<feature>* and **Use Case Design***<artefact>* loses **Conformity to target reality***<feature>* and then **Pilot One***<artefact>* loses **Conformity to target reality***<feature>*.

Risk scenario (natural language): *Business modelling is skewed by local viewpoints and results in missed target reality of the pilot implementation of the system.*

The risks were further identified by comparing the analyzed model with the Rational Unified Process (RUP) [11] taken as a referential model. RUP was particularly chosen as being well structured, defined in detail yet generally applicable and finally compatible with the development process of the studied project. Due to the limited resources for the case study, a complete mapping of the analyzed model on the RUP referential model was not developed. Instead, the most evident differences were taken into consideration. This way, three additional risk factors were identified. One of them is given below together with its exemplary scenario.

Risk factor: *Configuration & Change Management is not explicitly defined*

Risk scenario (in terms of risk patterns): If **Configuration & Change Management**<*activity*> is not performed then **System Integration**<*activity*> loses **Keep the set of integrated subsystems coherent**<*practice*> and then **Pilot Deployment**<*activity*> takes more time than expected.

Risk scenario (natural language): *Without explicitly defined change management process the pilot may not be integrated and deployed on time.*

The risk identification step was completed by comparing the analyzed model with the referential model derived from the Steve McConnell's 'Complete List of Schedule Risks' [5]. Following the same procedure as in the previous step, four additional risk factors were identified. One of them is given below together with the exemplary scenario.

Risk factor: *Long duration of the project and mainly part-time employment of the staff.*

Risk scenario (in terms of risk patterns): If **Project**<*activity*> loses **Maintain personnel continuity**<*practice*> then **Project**<*activity*> loses **Personnel**<*role*> and then **Project**<*activity*> loses **Avoid excessive schedule pressure**<*practice*> and **Pilot One**<*artefact*> loses **Completeness**<*feature*>.

Risk scenario (natural language): *The project can have difficulties with keeping part-time employed staff resulting in workforce shortages, more effort for available staff and limited scope of the pilot.*

In total, 11 risk factors were identified in Phase 1 of the case study. The results were then compared with the 9 risk factors indicated in the project description (identified by the project management). 6 out of the 11 risk factors identified in Phase 1 of our case study were also indicated in the project description. Still 5 of them were new regarding the project description and resulted in important suggestions for the process improvement which otherwise would have been missing.

Partial correlation of the detected risk factors with the factors indicated earlier by the project management confirms that the proposed method is consistent with the intuition and experience of the managers of software projects. The 5 new risks were then communicated to the Project Management Board who judged them important and initiated activities aiming at the process improvement.

The process was improved by defining its deficient areas in detail in the newly issued documents and initiating new activities related to the redefined process. The improvements covered in particular configuration management tools and practices, and procedures for quality assurance. All of the 5 deficient process areas identified with the help of the proposed framework were subject to the process improvements.

2.3 Results of Phase 2

The second risk identification and improvement attempt focused on the managerial issues such as operational management, quality management, communication management, software management and so on. A partial model of those areas was built from the available data. The model comprised 85 activities at 3 levels of detail, 16 roles and 37 artefacts.

Due to incompleteness of the process model the risk identification technique using model metrics could not be effective. Instead, the technique of comparison with a referential model was applied. For the same reasons as in Phase 1, the Rational Unified Process (RUP) [11] was selected as a primary referential model.

As a result, some 26 risks were identified suggesting possible process deficiencies. The examples of identified risk factors are given below together with possible scenarios.

- Quality management – establishing success criteria and metrics

Risk factor: *Immeasurable success criteria, weak metrics*

Risk scenario (in terms of risk patterns): If **Measurement Plan**<*activity*> loses **Use quantifiable and objective metrics**<*practice*> then **Pilot One**<*artefact*> loses **Defined scope**<*feature*>and then **Pilot One**<*artefact*> loses **Completeness**<*feature*>.

Risk scenario (natural language): *Product scope is arbitrarily redefined and intended product scope is not* *covered.*

- Document Management – general risks

Risk factor: *No explicitly defined procedures for maintaining traceability of key business and design decisions*

Risk scenario (in terms of risk patterns): If **Document Management**<*activity*> loses **Maintain traceability of key decisions**<*practice*> then **Documentation**<*artefact*> loses **Consistency**<*feature*> and then **Subsystem**<*artefact*> loses **Compatibility**<*feature*>.

Risk scenario (natural language): *Key business and design decisions are vague, inconsistent or contradictory in different documents resulting in incompatibility of partial commitments.*

- Communication Management – communication by project portal

Risk factor: *Poor performance of hardware and software platform of project portal*

Risk scenario (in terms of risk patterns): If **Project portal**<*artefact*> loses **Platform performance**<*feature*> then **Personnel**<*role*> loses **Motivation**<*capability*> and **Communication**<*activity*> loses **Follow defined communication paths**<*feature*> and then **Communication**<*activity*> takes more time than expected.

Risk scenario (natural language): *Discomfort in portal usage causes users' rejection and hasty construction of alternative communication means that impacts communication.*

The identified risk factors were taken as input to the risk analysis session carried out by the Project Management Board - PMB (some risk factors were merged together which resulted in total of 20 risks). The list of risks was also extended with the risks identified independently by PMB members. The risk analysis session involved rating and prioritizing the identified risks by some 15 members of the PMB. As a result, a list of project top 10 most important risks was elaborated. Of those 10 risks 7 were identified with the help of the proposed framework.

The risks found in this phase indicated more detailed deficiencies than those identified in Phase 1 and suggested areas for further process improvement. To save project resources only the areas indicated by 10 topmost risks were taken care of. For instance, some new actions were defined and launched in the area of technology management including tools selection and design decisions and communication means were defined in response to risks related to poor sharing of scientific know-how.

2.4 Summary

All the analytical tasks in the case study were carried out by one of the authors. However, the key part (risk identification) was performed according to precisely defined procedures of the proprietary risk identification techniques. Thus it may be presumed that every risk analyst would obtain comparable results. However, a complete correlation (reproduction) of the results is difficult to achieve, because while being considerably formalized the proposed method still engages the analytical capabilities and the individual perception of the analyst.

An effort on every phase of the case study was carefully measured. Phase 1 required 30 man-hours, 22 of which were spent on initial building of the process model while risk identification took 2-3 hours for each of the techniques and referential models. Phase 2 took 8 hours, 3 of them used to model the managerial area of the project and the remaining 5 to identify risks and potential improvements. The effort of the risk identification steps may be reduced by taking advantage of the already built dedicated supporting tool [8], which is currently adapted to the proposed framework.

3 Conclusions

The paper presented a two phase case study aimed at assessing the feasibility, effectiveness and efficiency of the risk based approach to improvement of a real-life software project. In the first phase, 11 risk factors were identified with the help of two different risk identification techniques. 5 of the risks were not known to the project management. All identified risks provided valuable suggestions for process improvement. In effect the process was significantly redefined. The repeated application of the approach in the second phase led to identification of some 26 more detailed risks and resulted in further process improvements.

The results of the case study demonstrate that the proposed framework is able to reveal new, previously undetected risks that provide important (in the opinion of project managers) input for process improvement. The framework requires that the process is defined in the level of detail sufficient to build its model which is then used during risk identification. A high-quality relevant referential model is also necessary if the model comparison technique is to be used. As the proposed approach is based on models and their metrics we expect the delivered results to be highly independent of the analyst's intuition and experience. It also provides a good base for automatic tool support.

Further research is planned in the following areas:

- further case studies with industrial partners,
- investigation of new metrics to improve risk identification,
- building the proposed framework into a supporting tool [8],
- using the framework in different domains such as e-health and/or e-commerce.

The results presented in this paper originate from a wider context of research towards a holistic approach to risk identification and process improvement in software projects supported by dedicated tools. The description and the results of the research are available at [12].

4 Literature

1. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, V1.1, Tech. Rep. CMU/SEI-2002-TR-004, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 2002
2. Jaccheri M.L., Conradi R., Techniques for Process Model Evolution in EPOS, IEEE Trans. on Soft. Eng., Vol. 19, No. 12, 1993, pp. 1145-1156.
3. Jones C., Assessment and Control of Software Risks, Yourdon Press, New Jersey, 1994.
4. Kontio J., Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation, PhD Thesis, Helsinki University of Technology, Finland, 2001.
5. McConnell S., Rapid Development, Microsoft Press, 1996.
6. Miler J., Górski J., Risk Identification Patterns for Software Projects, Found. of Comp. and Dec. Sci., Vol. 29, No. 1-2, 2004, pp. 115-131
7. Miler J., Górski J., Identification of Software Project Risk Based on Process Model, accepted for VI National Conference on Software Engineering, Gdansk, 2004 (in Polish)
8. Miler J., Górski J., An environment supporting risk management in software projects, proc. of I National Conference on Information Technologies, Gdansk, 2003 (in Polish)
9. Software Process Engineering Metamodel Specification Version 1.0, Object Management Group, November 2002.
10. PMBOK Guide, 2000 Edition, Project Management Institute, 2000.
11. Rational Unified Process version 2002.05.00.25, http://www.rational.com/rup
12. RiskGuide project website, http://mkzlway.eti.pg.gda.pl/RiskGuide
13. Sisti F. J., Joseph S., Software Risk Evaluation Method, Tech. Rep. CMU/SEI-94-TR-19, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 1994.

5 Author CVs

Jakub Miler

He is a PhD student at the Department of Software Engineering, Gdansk University of Technology. His primary research area is software risk management. Under the supervision of Janusz Górski he prepares the PhD thesis titled "A method of identification and analysis of risk in software projects". Recognizing the risk management as an effective method for software process improvement, he works on useful techniques for risk identification and analysis together with a model of risk representation and classification. He also investigates the methods of communication of risk-related information within a project team and develops the RiskGuide software tool. In 2000 he presented the master's dissertation titled "Computer system for supporting risk management in a software engineering project".

Janusz Górski

He is a professor and the Head of the Department of Software Engineering at the Gdansk University of Technology. Authored and co-authored 6 books and more than 150 research papers. Led several software development projects in the domains of process control, telecommunications and information management. His present interests include software and system engineering and risk and trust management. In those areas he actively carries research (through a number of EU funded and national projects) as well as cooperates with industry through common projects and consulting. In 1999 he initiated the National Conference on Software Engineering which is since then organized every year.