DRAFT

full paper published in:

proc. of 4th National Conference on Software Engineering
October 15-18, 2002 Poznań - Tarnowo Podgórne, Poland

Paper published in the proceedings and presented at the conference

Supporting Team Risk Management in Software Procurement and Development Projects

Jakub Miler, Janusz Górski
Department of Applied Informatics
Technical University of Gdańsk
Gdańsk, Poland

Abstract

This paper presents a method of collaborative risk management in software procurement and development projects. First it overviews the success criteria and typical risks from the point of view of different stakeholders. Then a concept of the continuous process of risk identification and analysis is presented together with the structures supporting the management of the process and the representation of risks. We also propose a security policy to be applied in order to reflect possibly conflicting interests of project participants. Finally, a practical solution – the software tool supporting risk assessment, together with the results of three validation experiments are briefly presented.

1. Introduction

Software projects still face serious problems throughout their life and many of them miss their goals in terms of time, budget and user satisfaction. It is estimated that some 70% of projects still have problems with reaching their objectives [12]. Present software projects are facing constantly changing requirements, drifting project scope and are put under

demanding schedule pressure and budgetary constraints. On a technical level, software development becomes more challenging as the size and complexity of the systems being developed is growing. On a managerial level, the project management tends to be more complicated as the project constraints are more tough, project size increases and projects often involve several different participating organizations.

The success of a project is typically assessed by the degree to which the project meets the time and budget constraints and, first of all, by the achieved level of user (customer) satisfaction. The risk of a project failure relates to the possibility that the project steers away from its success space. Risk management has been recognized as the method to keep a project on its way to the success, e.g. in [9] it is listed among nine key knowledge areas related to project management. In the recent years risk management receives much attention in the software engineering community (see e.g. [1, 3, 4, 5, 6, 10, 11]). A project with risk management aims at early identification and recognition of risks and then actively changes the course of actions to mitigate and reduce the risk. This requires open communication, forward-looking attitude, team involvement and access to the knowledge base of typical problems.

The overall success of a project depends to a great extent on effective co-operation of all stakeholders, first of all the developers and the users/customers, but also suppliers, maintenance engineers, external consultants etc.

Many risks in a project are relevant to several stakeholders and their identification and analysis can be amplified if the stakeholders are encouraged to co-operate. This can be facilitated if the stakeholders have access to a broad and continuously open communication channel through which they can pass the risk-related information.

Concerning risks, however, the set of project stakeholders is not uniform and this should be reflected in the structure of the risk management process. For instance, the stakeholders can differ in the phase of the project they are involved in (e.g. analysis, design, testing) or in the scope of their view of the project (e.g. top managers, managers, engineers). What is less obvious is that in some cases the objectives of the stakeholders (or groups of stakeholders) can be partially in conflict and therefore their views have to be subjected to some security constraints. As an example, let us take the managers' view from the supplier and the client sides. Some risks on the client side (e.g. financial problems considered as "temporary") probably will not be communicated to the supplier, even if they relate to the common project. Such risks will be kept confidential within the organization and will not be released to the outside. As the information on risks often has strategic value, some security policies should be applied reflecting the interests of the co-operating organizations and the interests of the project as a whole.

The objective of this paper is to present a concept of the risk management process that encompasses two co-operating organizations: the supplier (who runs a *software development project*) and the customer (who runs the *software procurement project*). The two projects are not completely separated and independent, however. Instead, they share common risks and their success depends on each other. We propose a continuous process of risk identification and analysis that can be applied in each participating organization. This process is facilitated by a broad and highly available communication channel through which the project team members can communicate risk-related information. We

also define the interface between the processes through which they can exchange the risk related information in accordance with the corresponding security policies (see Fig.1).

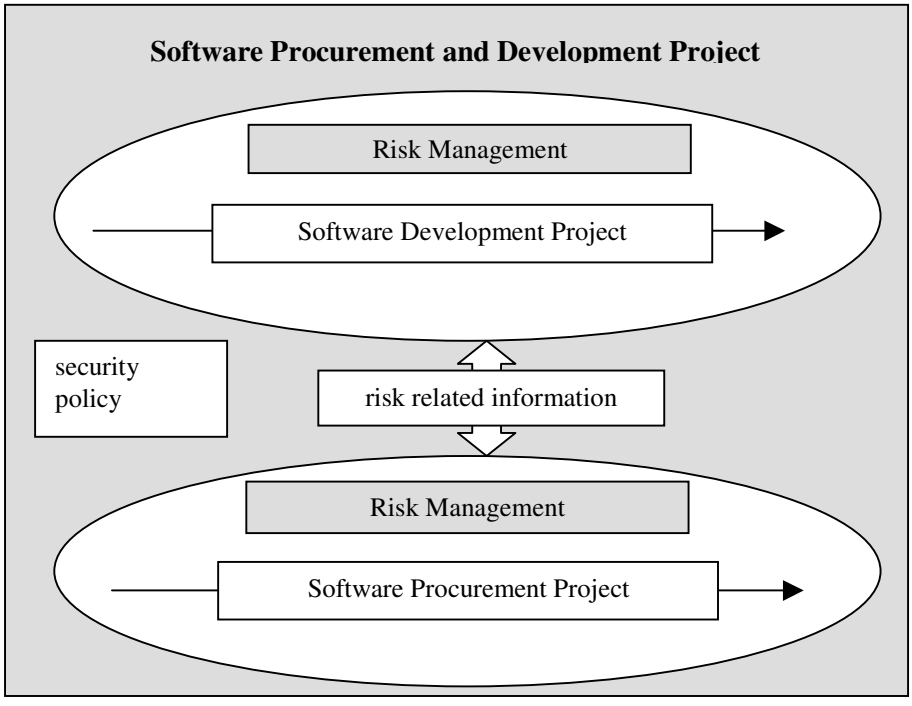

Figure 1: Risk management during software development and procurement

In the subsequent sections, we overview the success criteria and typical risks from a point of view of different stakeholders and introduce the risk assessment process. We also present a software environment supporting the risk identification and communication together with results of experiments performed in order to validate our approach.

2. Risk in software projects

Risk relates to the possibility to degrade the success of an undertaking. The success is defined by a set of criteria that the outcome or the solution must meet to be considered "successful".

The overall success criteria for most of projects include:

- reaching the adequate functionality and quality that provides for achieving the business objectives of the system,
- finishing the project on time,
- keeping the expenses within the budget,
- achieving sustained customer satisfaction.

In terms of a business outcome, the success criteria can for example mean one or a combination of:

- increasing the market share,
- outrunning the competitors,
- improving the effectiveness or productivity,
- cost reduction,
- capturing a particular segment of the market.

Steering away from these goals results in a diminution of the overall project success. The success criteria are often formulated differently from a point of view of each project stakeholder and it is important that they converge and do not knock the project out of its success space.

For a project as a whole, some of the most important risks and their exemplary consequences include:

- lack of user satisfaction – canceling a project, product rejection;
- poor product quality – high maintenance costs, product rejection, adversely affecting the developer reputation;
- missing the deadline – loss of business opportunity, troublesome social and political atmosphere;
- overrunning the budget – financial losses, project cancellation.

From the developer's point of view there are some risks that reflect its particular concerns, like:

- lack of financial stability at the client, delays in payments,
- the client cannot be "convinced" that the product is satisfactory,
- inquisitive, nosy and demanding client.

On the other hand, there are also some risks that are related to the concerns that are specific for the customer, like:

- supplier is unreliable and does not fulfill its commitments,
- exaggerated financial demands of the supplier,
- the product is useless although meets most of the requirements,
- situation changes and there is no more money to finance (otherwise successful) project.

If we take a closer look on the above we can see that the risks perceived from the customer and developer perspectives can sometimes reflect particular interests of the participating institutions that will not necessarily be willing to share them with the other side. This seem to be in conflict with the general principle of information sharing and team approach to risk management. A right approach seems to be to apply a sort of "filter" that imposes some restrictions on otherwise free information exchange during risk identification and assessment activities. Such a filter can be implemented by a dedicated security policy (see Fig.1). The proposed risk assessment process and security policy are described in the subsequent sections.

3. Risk assessment

Our risk assessment is based on three concepts: *reviews*, *snapshots* and *reports* that underpin the three layers of processing the risk-related information: *identification*, *analysis* and *reporting* [2]. Reviews establish the framework for risk identification. Snapshots pass the identified risks for further analysis. Reports communicate the results of risk assessment.

The risk identification layer uses reviews to gather risk-related information from a project. Reviews differ in terms of their scope, duration, participants and identification techniques. It is possible that two reviews overlap in time, however differing in their scope and/or participants. Risk-related information collected during a review is represented as *risk indication* and identifies a particular risk, the involved project stakeholder, timestamp, the identification technique and possible comments.

For any defined period we define *risk snapshot* as a report showing, in a predefined form, all the risks identified during this period. Thus, a snapshot is a sort of the "map of identified risks" with removed redundancies and timing information. A snapshot presents to the risk manager the present state of risks (in terms of risk indications). This provides an insight into the information collected during a review and may be used to decide on closing the review and passing to the assessment phase.

Risk snapshots form the input to the risk analysis. The analysis is summarized in the risk assessment report. The report contains the concise and agreed upon information on risks in the project. It can then be used as input for risk mitigation related activities. It may also be taken as an input to the next risk review action. The cycle of risk identification and analysis is shown in Fig. 2.

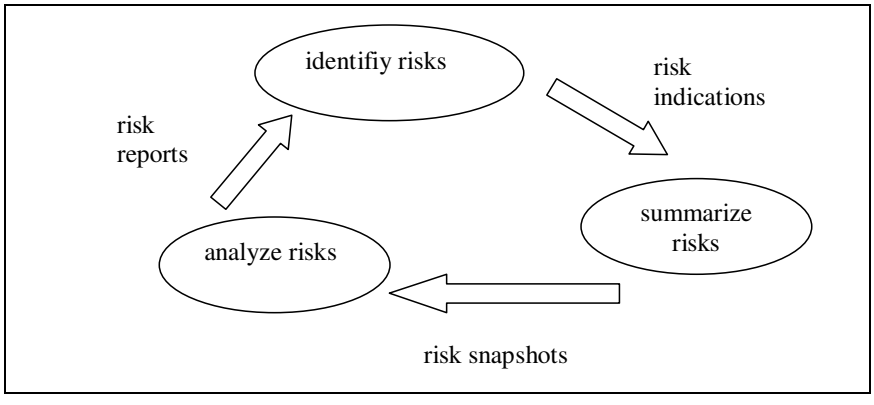

Figure 2: The risk identification and analysis cycle during risk reviews

4. The process

Opening a channel to communicate and memorize risk related information is not enough, as it does not guarantee that anything is actually communicated and memorized. Causing that the information is actually generated is the manager's task. We assume that there is a risk identification and analysis process performed by the project stakeholders and controlled by the risk manager (the role usually played by the project manager except large projects where it could be assigned separately). The process is structured as a sequence of reviews as is shown in Fig. 3.

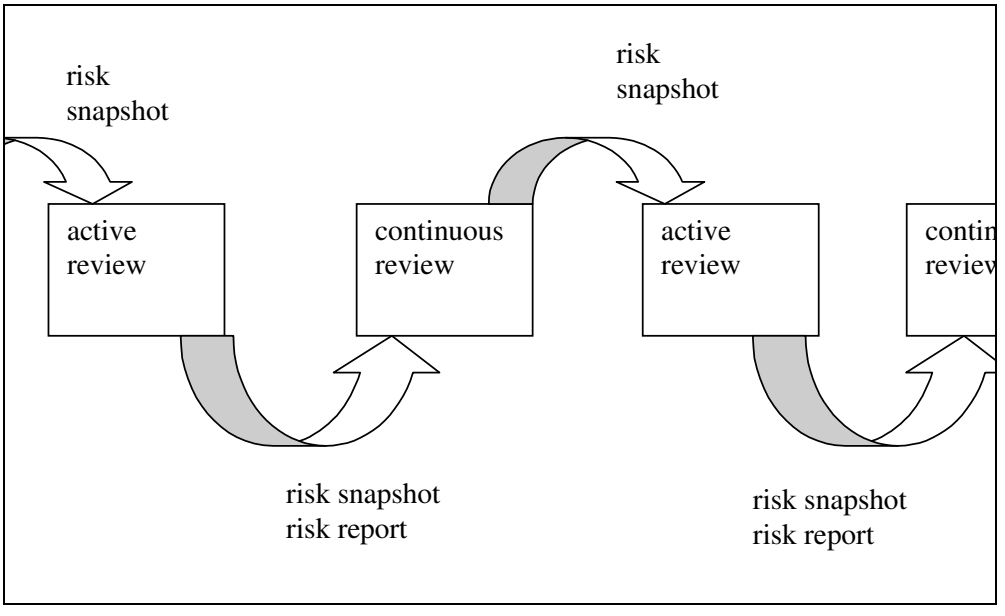

Figure 3: The continuous risk identification and analysis process. The arrows represent new information on risks generated by the process

It is assumed that at any time at least one review is open. The review remains open over its time window. Time windows of subsequent reviews are adjacent. We distinguish between two types of reviews:

- active review – its starting and ending times are set by the risk manager as well as its scope and participants (the stakeholders involved in the review). The review has a defined set of inputs (reports, checklists, questionnaires, etc.) and associated risk identification techniques. As a rule, the snapshot from the last continuous review, is included as an input of the active review. The active review ends with the risk analysis session that aims at assessing and prioritizing the identified risks and produces a relevant report.
- continuous review – it starts with the end of the previous review and ends with the start of the next review (being it active or continuous). It just keeps the communication channel open enabling the communicated risk information being memorized. The set of its input documents is not controlled by the risk manager.

Any project stakeholder can pass risk-related information disregarding the way of its generation.

Typically, a snapshot is taken at the end of the continuous review to provide an input to the subsequent active review. A snapshot is also taken at the end of an active review to summarize the effects of risk identification activities (as shown in Fig.3.). Additionally, a formal risk assessment report is generated at the end of an active review.

We assume that the process has the active and continuous reviews interleaved, their extent (in time) and scope (in terms of inputs and participants) being controlled by the risk manager. This way we achieve the following benefits:

- the communication channel is constantly open,
- the identification actions are being planned (active and continuous reviews),
- all communicated risk-related information is being memorized,
- the identified risks are periodically reviewed and assessed and the frequency and scope of those assessments is under control of the risk manager,
- the results of the analyses are kept in the form of reports and are available downstream of the process (can support further identification and analysis).

5. Collaborative risk assessment

As it is shown in Fig.1 the development and procurement projects share the risk related information to increase the chances of the final success. However, in section 2 we pointed out that it is rather unlikely that the parties (supplier and customer) are always willing to exchange the risk related information in a free way. We propose risk reports as the tokens being exchanged to communicate information about risks. This means that the parties (supplier and customer) perform their own individual risk assessment processes as shown in Fig. 3. The processes are synchronized on points in time when they exchange their formal risk reports (Fig. 4).

Figure 4: Parallel processes exchange risk assessment reports

At the initiation phase of their projects, both parties launch their risk management processes. After a party completes the analysis and publishes the report, it can decide to make it accessible to the other party. This way the results of risk analysis on one side can be taken as an input to the risk management process on the other side. The

communication channel between the parties remains open and accessible, but offers some means to control the information passed outside the participating organizations.

The explicit separation of risk assessment processes of the participating organizations provides for maintaining the autonomy in managing the individual process and controlling the communication between them.

6. Representation of risks

The risk management process should be supported by adequately defined data structures maintaining the risk-related information. Our model of risk representation is shown in Fig.5.

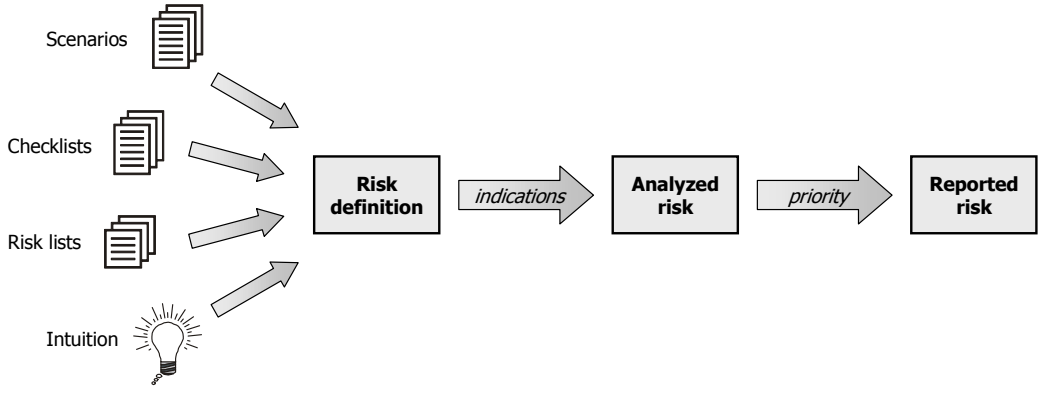

Figure 5: Risk representations

Risk definition can be a simple statement expressed in a natural language, a formal expression in a certain notation or a scenario showing how we can get to an undesired state/event. In our present process, risk definitions primarily come from the predefined checklists and lists of risks, but we also recognize the need of defining new situation-specific risks that are recognized by project stakeholders using intuition and engineering judgement.

If a stakeholder recognizes a given risk as being present in the project, it creates an *indication* of this risk. Multiple stakeholders may indicate the same risk as well as it may be indicated many times by the same stakeholder (e.g. in different reviews).

The indicated risks are collected in snapshots (as described in the previous sections) and mapped into the *analyzed risks.* An analyzed risk comprises additional information like: priority, evaluation of its likelihood etc. It is important to provide for backward and forward traceability between the various risk representations to provide for visibility of the analyses and retrospection. The most important analyzed risks are selected for publication in the risk assessment report. A *reported risk* is the analyzed risk extended with a risk response related information, e.g. contingency plans.

7. Security policy

The information on identified risks often has strategic value for an organization as it can reveal some drawbacks, conflicts, negligence or even symptoms of upcoming crisis. Releasing of such information to the outside is usually under some security controls and it is unlikely that the organization completely relaxes those controls for the sake of a common project. Thus, the appropriate mechanisms that guarantee the necessary confidentiality should be in place. The mechanisms should implement the agreed security policy governing the access to risk related information inside and between the participating organizations.

We suggest that the access to the risk related information is controlled by a mandatory security policy [13]. Such a policy considers information assets (objects) and accessing subjects (in our case, the stakeholders on both sides: supplier and customer). A classification hierarchy is assumed for both, objects and subjects. Each object and every subject are assigned a confidentiality level, denoted as *CL*, which has two components: (1) classification *C* which, for example, can be one of: *unclassified*, *confidential* or *highly confidential* (different security policies can define different confidentiality classifications) (2) level *L* from a set of distinguished organizational levels (e.g., senior manager, manager, engineer etc.) which the subject or object is associated with. For example, an employee's *CL* can be {confidential; manager}. A confidentiality level CL_1 is said to *dominate* another CL_2 if and only if: $CL_1.C \geq CL_2.C$ and $CL_1.L \geq CL_2.L$. Confidentiality requires that confidential information is not exposed to less confidential subjects. The following two access rules protect confidentiality of information:

R1: A subject *S* can read an object O if and only if CL_S dominates CL_O i.e., the subject can read-down or read-equal, but can never read-up.

R2: A subject *S* can write an object *O* if and only if CL_O dominates CL_S i.e., the subject can write-up or write-equal, but can never write-down.

Rule R1 alone is not sufficient to ensure confidentiality and rule R2 ensures that the contents of a confidential object is not indirectly exposed to a an unclassified subject: when a confidential subject writes-down into an unclassified object the information which he read from a confidential object, confidentiality is violated if the unclassified object is permitted (by rule R1) to be read by an unclassified subject. So, rule R2 is necessary to prohibit write-down.

The overall concept of the security policy is presented in Fig. 6.

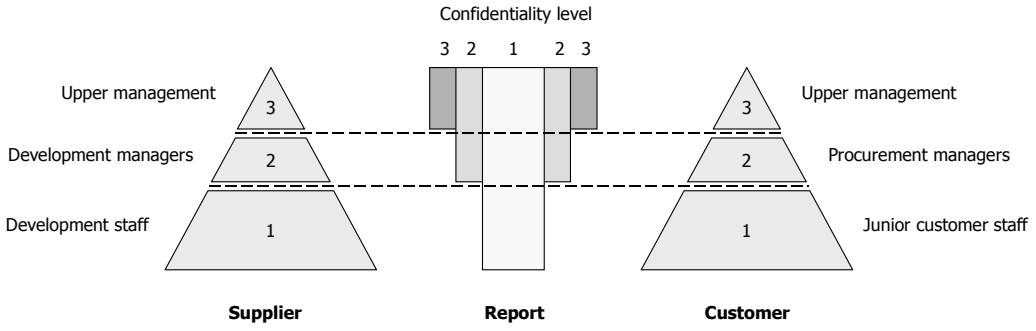

Figure 6: The proposed mandatory security policy

In our approach, all the participants of the project representing the collaborating organizations are assigned to a confidentiality level (presently we distinguish three levels: senior manager, manager, staff). Then, in the risk identification phase, each risk is classified based on the classification of the stakeholder who identified it or higher. This classification can then be changed during the risk analysis phase. The analysis is performed separately in each participating organization. In the risk report, each reported risk (see Fig. 5) has a confidentiality level assigned that restricts the group of recipients only to the project members with the domination security classification.

In addition to the mandatory security policy described above, a discretionary security policy is being applied within each classification level while risks are communicated between the participating organizations. Each risk included in the risk report that is made accessible to the partner organization can be access restricted and consequently made invisible on the other side.

8. A practical solution – the Risk Guide system

To carry out experiments validating the concepts proposed above, we have designed and implemented a software tool ❮❯Risk Guide supporting the risk management process [7]. It is installed on a central server and accessed by the users simply with a web browser. This architecture makes the system applicable in distributed software projects, where the stakeholders cooperate from different geographical locations. In addition, the tool can be used in multiple projects at a time and support many independent risk identification and assessment processes. A demo of ❮❯Risk Guide can be found on our web site [8].

The system comprises a knowledge base of typical risks together with the questionnaires to help in thorough risk identification. As for now, the ❮❯Risk Guide knowledge base is built of two components: the public knowledge base that contains freely available checklist and list of risks, and the restricted knowledge base that includes a more advanced proprietary questionnaire and the related list of risks. This latter questionnaire comprises some 404 questions that cover the following areas of a project: project type and size; contract; upper management support; project planning; collaboration with the customer; target system and its environment; system development process; system design

and implementation; configuration management; quality management; personnel management.

The tool supports the management of checklists and lists of risks as well as provides for the evolution of risk definitions (by means of versioning).

Risk Guide supports full risk assessment process as presented in Fig. 3 implementing the proposed concepts of risk reviews, indications, snapshots and reports. Before the process starts, it is necessary to register the accounts for all stakeholders participating in risk management. Done with this phase, the risk manager can open the first review to initiate the risk identification phase. The tool records all risk indications submitted by project members. Presently, Risk Guide supports the following risk identification techniques:

- automatic generation of risk indications based on the answers to a questionnaire,
- explicit selection of a risk from the list of risks,
- supplying a new definition of a specific risk (e.g. identified by intuition and/or engineering judgement) and then referring to it,
- referring to the previously identified risks.

To start the analysis phase, the risk manager takes a snapshot of already identified risks. Once the snapshot has been taken, all open reviews are closed and further identification requires opening a new review. The analysis is carried out in two steps: evaluation prioritizing of risks. Risks are evaluated in three dimensions: possibility, severity and timeframe. For each dimension Risk Guide offers a qualitative evaluation scale. An overall risk evaluation is derived from the individual evaluations using the risk evaluation matrix embedded in the system. In addition, the analyzed risk includes a comment that can be added to justify its evaluation it or to express certainty of this evaluation. The priorities can be assigned automatically (deriving them from the information contained in risk evaluation) or by the manager's decision. The list of evaluated and prioritized risks is ordered, so the most important risks are available on top of the list.

Once the analysis is completed, the resulting list of the topmost risks is published in the risk assessment report. By definition this report is available for the registered stakeholders. It can be also made available to another team and this way the information about risks can be exchanged between otherwise independent risk management processes. An example of a risk assessment report is presented in Fig. 7.

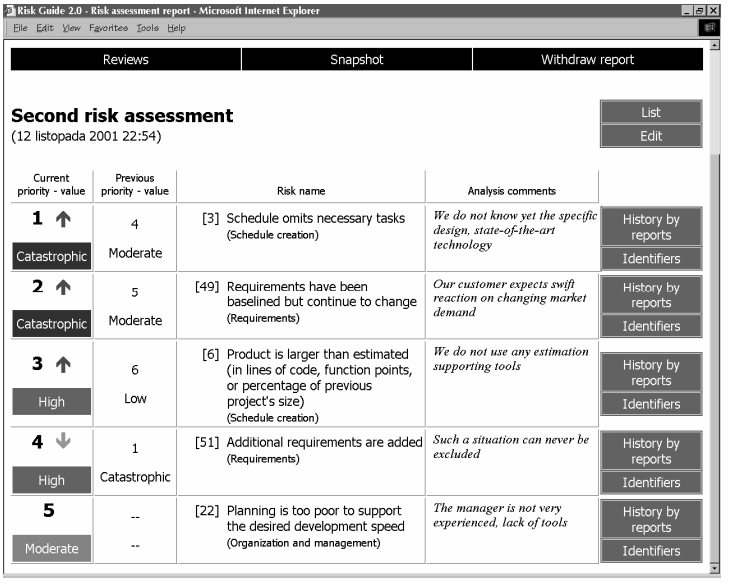

Figure 7: Example of a risk assessment report

The tool also offers various options supporting risk tracking and risk history analysis.

As stated earlier in the paper, the risk information gathered in the process and published in the reports often has strategic value and should be protected. Presently Risk Guide incorporates two types of security and access control mechanisms:

- functional – explicit separation of roles of risk manager and project member, and access to external reports restricted at a level of a particular risk,
- technical – ciphering of Internet connection using Secure Socket Layer 3 (SSL3), password based access to all system components and data.

9. The experiments

On our way towards an integrated environment for team risk management we plan and carry out experiments to validate our concepts. As for May 2002, two experiments are already completed and the third is now being evaluated.

The first experiment was carried out in the academic year 2000/2001 during the Software Engineering Project Management course at Technical University of Gdańsk [7]. It aimed at the evaluation of the ease of use of Risk Guide and the assessment of the effectiveness of its support in a single risk assessment cycle (according to Fig. 2). Altogether, 38 student groups participated in the experiment and identified the risk in their software projects, either fictional or factual. The identified risks were then taken into account while building the detailed project plan. Finally, the experiment was evaluated using a questionnaire with answers in the scale 1-5 and the calculated average of the answers formed the results. Detailed results of the experiment are available in [7]. The results confirmed the value of providing a common environment to focus the participants'

attention on risk management and provided numerous suggestions of user interface improvements.

The second experiment took place in the period October 2001 – January 2002 and was carried out during an industrial training course in software engineering [2]. It aimed at the evaluation of the concepts of snapshots and reports as well as assessing the effectiveness of support in relation to schedule risks. Full recurring assessment process (as presented in Fig. 2) has been followed. Two small projects were involved in the experiment (each of 5 team members), both with the same goal – the development of a billing system for a telecom switch. The participants were software engineers coming from local companies. The results were assessed by a detailed questionnaire at the end of the project as well as the overview of the manager's reports and the risk management history recorded in Risk Guide. The experiment confirmed the help of the tool in increasing team awareness of risks and focusing at particular risk areas. The knowledge base of Risk Guide offered to the participants was highly evaluated. Statistical results could not be calculated because of very small sample.

Finally, to test our concepts in the environment of a real-life project, we plan a series of experiments with software companies and customer organizations that could incorporate the Risk Guide tool into their projects. We offer to our partners the access by the Internet to the system installed at our labs. Together with it we offer technical support in the use of the tool and in the organization of the risk management process. The form of collaboration during such experiments is presented in Fig. 8.

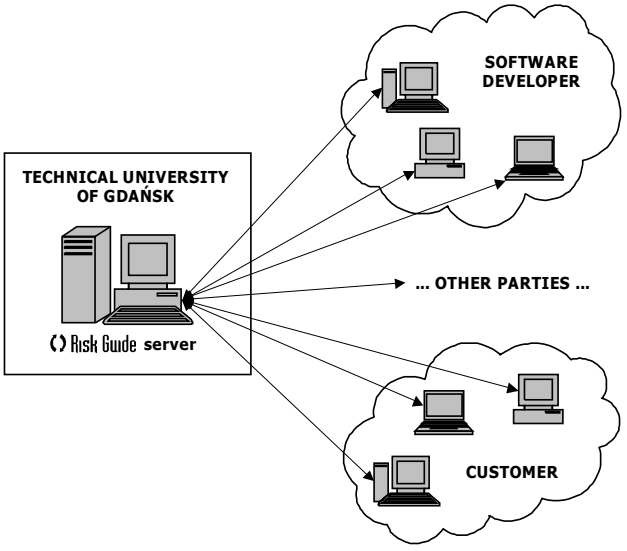

Figure 8: Experiment organization

The experiment commences with a preparatory meeting during which the objectives and scope of the experiment are discussed, the role of each contributing organization is explained, the supporting tool is demonstrated and the first brainstorming session of risk identification is being performed. The meeting also includes the appointment of the risk manager in each risk management process related to the participating organizations. Next,

the organizations running the software project carry out the risk assessment cycles ending in a risk analysis session. Throughout the experiment various data are being collected to provide for its assessment.

The first experiment of the series started in February 2002. It was intended to help assessing the support offered by ()Risk Guide in more quantified terms as well as to evaluate and enhance the knowledge base. In opposition to the first two experiments, which were using the open knowledge base of ()Risk Guide, this experiment uses the proprietary knowledge base. It was related to a real project of development of a GIS system to support management of a heat generation and distribution infrastructure in one of the major cities of Poland. The plan of the experiment presumed performing several cycles of a continuous process and the elaboration of several risk assessment reports. Moreover, we expected the participants to contribute to the ()Risk Guide knowledge base and supply their own specific risks. In the course of the experiment, due to the changing conditions of the real-life project, the experiment was limited to the first identification attempt. Nevertheless, some interesting opinions were gathered from the participants and in the end the experiment resulted in very useful lessons learnt:

- the particular interests of stakeholders of the risk management process must be preserved,
- appropriate access control policies must be embedded into the process,
- the knowledge base should be better structured and contain more specific risks,
- the risks related to software process improvement should be distinguished,
- sometimes it is convenient to include a preliminary evaluation in risk indication,
- the built-in lists of risks should be better structured and the risks identified by answering a chapter of a questionnaire should be reviewed and tailored by the identifier.

10. Conclusions

In the paper we proposed a process of collaborative risk assessment that comprises risk assessment processes to be applied within each participating organization and a method of sharing the risk-related information between the processes. To make it effective we proposed a security policy to be applied to control access to the risk-related information.

Risk management can benefit from tools that support communication and collaboration. We described an Internet tool offered to project stakeholders disregarding their actual geographic dislocation. The tool supports the risk assessment process internal to a particular organization as well as sharing the reports with other processes ensuring appropriate access control.

Finally, we presented the series of experiments carried out to validate our concepts in practice. These experiments confirmed the value of providing an integrated environment with a knowledge base of typical risks to the project participants to focus their attention on risk management and guide the risk assessment process. The first two experiments were limited to a single software development project and did not include the

participation of the customer representation. In this case the common risk repository was considered very useful. However, the last experiment has led us to the conclusion that in a real-life software project some areas are strictly related to the strategic interests of the stakeholders and should be kept confidential. Based on these results we have improved our model as presented in this paper to provide for independent but coordinated risk management processes that exchange information obeying a defined security policy.

A summary of our research, the articles, the presentation of Risk Guide risk management tool together with the latest reports from experiments can be found on our web site [8].

References

[1] Galagher B. P., *Software Acquisition Risk Management Key Process Area (KPA) – A Guidebook Version 1.02*, SEI report CMU/SEI-99-HB-001, Carnegie Mellon University, Pittsburgh PA, October 1999.

[2] Górski J, Miler J., *Towards an integrated environment for risk management in distributed software projects*, Proc. of 7th European Conference on Software Quality, Finland, 2002.

[3] Higuera R. P., Gluch D. P., Dorofee A. J., Murphy R. L., Walker J. A., Williams R. C., *An Introduction to Team Risk Management*, SEI report CMU/SEI--94-SR-01, Carnegie Mellon University, Pittsburgh PA, May 1994.

[4] Ikeda H., *Building in Quality through Information: A Trial Use of the TIPS Technical Information System to Reduce Project Risks*, Proc. of 2nd World Congress for Software Quality, Japan, 2000.

[5] Jones C., *Assessment and Control of Software Risks*, Prentice Hall, 1994.

[6] Kontio J., *The Riskit Method for Software Risk Management, version 1.00*, Techn. Rep. CS-TR-3782, Department of Computer Science, University of Maryland, USA, 1997.

[7] Miler J., Górski J., *Implementing Risk Management in Software Projects*, Proc. of 3rd National Conference on Software Engineering, Poland, 2001.

[8] http://mkzlway.eti.pg.gda.pl/RiskGuide

[9] PMBOK Guide, 2000 Edition, Project Management Institute, 2000.

[10] Software Engineering Institute, http://www.sei.cmu.edu/

[11] Tanida T., Tsuruwaka H., Takahashi N., *Application of Risk Management Technique in Software Development*, Proc. of 2nd World Congress for Software Quality, Japan, 2000.

[12] http://www.softwaremag.com/archive/2001feb/CollaborativeMgt.html (visited 25.05.2002).

[13] Gasser M., *Building a secure computer system*, Van Nostrand Reinhold, 1988