Software Risk Management: Using the Automated Tools

Sergey M. Avdoshin, Elena Y. Pesotskaya

School of Software Engineering, Software Management Department, National Research University Higher School of Economics, Moscow, Russian Federation.

{Savdoshin, Epesotskaya}@hse.ru

Abstract. Software development process nowadays faces many challenges and risks. In order to manage risks we need to understand the scope and objectives of the software developments and use the appropriate automated risk management tool. The study addresses software risk management in software development area and an approach to analysis, structuring, and evaluating risk with the help of specialized automated tools. The author provides recommendations on how to define a set of selection criteria for automated tools and analyses the growing demand for service hosting solutions and web-applications, stressing that almost any software including risk management tools can be successfully run using this method.

Keywords: Software risks, risk management, risk management software, risk management tools, web-based applications, SaaS solution.

1 Introduction

There are many risks involved in creating high quality software that need to be carefully managed. Despite new technology, innovative methods and tools, different management methods - development process still full of risks from the beginning to the end. Therefore, to make sure a project successful we require managing specific IT risks related to our software projects: identify risks and store in a shared data storage, assess risks, using specialized tools and techniques, choose appropriate mitigation action and track that mitigated risks are lower when they were. The need for project risk management has been widely recognized by all software development companies such as Microsoft, SAP, Oracle, IBM etc. Being a development companies they usually use their own powerful automated tools to minimize losses and maximize software development success. As the purpose of project risk management is to improve project performance by systematically identifying and assessing risks, developing strategies to reduce or avoid risks, and maximizing opportunities [2], risk management tools should support a continuous risk management process throughout the life cycle of a system. Effective risk management depends on risk management planning; early identification and analyses of risks; early implementation of corrective actions; continuous monitoring and reassessment; communication, documentation,

and coordination. The science of risk management was developed back in the sixteenth century during the Renaissance, a period of discovery, but regarding the subject of Risk Management Process (RMP), since 1990 a large number of methodologies and methods have been generated to address the need for more effective risk management [7]. Among them we can distinguish the PUMA [5] and the MRMP [8] in construction engineering context; the RFRM [6] in system engineering context; the SHAMPU [2] and the PMBoK [9] in project management context; the standard of the AS/NZS 4360 [4] and the DoD [3] in public application context, etc. In the present paper, we have investigated and compared most of risk related topics in software engineering context and automated tools that support risk management process.

Risk management should begin at the earliest stages of program planning and continue throughout the total life-cycle of the program. Additionally, risk management is most effective if it is supported with automated tool that ensures integration with the program's systems engineering and program management processes.

Common practices, concerning risk management is to identify and track issues and risks and then manage the root causes or the consequences (if we were not successful while managing root causes). But also the objective of a proper risk management tools is to provide a repeatable process for balancing cost, schedule, and performance goals within program funding, especially on programs with designs that approach or exceed the state-of-the-art or have tightly constrained or optimistic cost, schedule, and performance goals. Successful risk management depends on the knowledge gleaned from assessments of all aspects of the program coupled with appropriate mitigations applied to the specific root causes and consequences.

Software risk management is not a stand-alone task. It is supported by a number of other tasks as the results of risk management are used to finalize requirements development, logical solution, systems engineering, cost estimating, schedule development, performance measurement, etc.

1.1 Process and tools for software risk management

In software development process, risk management concerns all aspects of the program life cycle phases as they relate to each other, from initiation to disposal. There are basic risks that are generic to almost all software projects. In reality many IT projects are very similar at a high, strategic level. They differ in people involved and exact events. An effective risk management process requires a commitment on the part of the project manager, the project team, and the contractor to be successful. The project team and management should establish a risk management process that includes not only risk planning, but also risk identification, risk analysis, risk mitigation planning, risk mitigation plan implementation, and risk tracking to be integrated and continuously applied throughout the whole program. For that purposed some automated risk management tools provide seamless integration with Microsoft® Project to quantify the cost and schedule uncertainty associated with project plans. Other tools have possibilities of integration with Microsoft® Excel and illustrate

many possible outcomes, e.g. cost and schedule histograms in your Microsoft® Excel spreadsheet.

In the planning phase, the goal of successful risk management is to adapt risk management to the organization's existing project and risk management practices and to document the resulting processes in a risk management plan. Any computerized tools, databases or forms are installed. Project personnel is being trained both in the processes to be carried out and in the methods and tools to use. Finally the risk management activities must be started and become habitual routine within the project.

Also before we start the software risk management, several questions should be answered:

- What do we expect from risk management?
- Who would participate, how often?
- What skills, competencies are required for risk management?
- What are the main risk management steps and deliverables?
- What actions would be conducted on each step?
- What instruments and methods should be applied at each step?
- What terminology do we use?
- What are the criteria for risk prioritization?
- What response actions should be taken for risk avoidance, mitigation?
- How we should monitor the risk response actions?
- What control activities we apply to the risk management process?
- How often we do reporting on risk management?
- What supporting tools do we use (database, software tools, metafiles, communication channels, etc.)

The automated tool selected for risk management purposes should be integrated with the risk management process methodology involves five basic steps [2], [9]:

1. Identify the risks - Understand the typical problems that might adversely affect the project.
2. Assess the risks - Rank the risks in order of importance based on probability of occurrence, impact of occurrence, and degree of risk certainty.
3. Plan the risk response – Analyze risk assessment alternatives and modify the project plan to adjust for the risk.
4. Monitor the risks – Throughout the project, continue to revisit the risk profile, re-evaluate major risks, and update the risk profile with action taken.
5. Document lessons learned – Learn from the risk identification, assessment, and management process.

During the first step in the software risk management process, risks should be identified by the project team and interested parties and added to the list of known risks. The automated tool usually supports many techniques for identifying risks, including interviewing questionnaires, reporting, decomposition, assumption analysis, critical path analysis, SWOT, etc. Identifying software risks involves collecting information about the software development project and classifying it to determine the amount of potential risk to the project. Identification procedures include as many participants as possible: team members, experts, functional departments, sponsor, end users, other interested parties. It does not mean all the interested parties need the access to the risk management tool. It depend on the risk management process

organization how the information would get to the system. As an output for risk identification stage – risk register should appear (fig.1 shows an example).

#	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	Risk name	Description	Category	Root cause	Consequence	Triggers	Risk owner	Status

Fig. 1. Risk register sample

Sometimes Risk Register contains some extended fields that should be filled in the later stages of risk management, e.g:

- Risk responsible - person who will take responsibility for each risk (might be the same as risk owner);
- A rank for risk as a result of risk prioritization;
- Potential responses to each risk;
- The probability and impact of each risk occurring.

The successful Project Manager should lead a discussion amongst the team and sponsors to determine the high, medium, and low categories based on the Risk Scores, and assign responses to those categories.

The value of software tool is increased if there are software checklists available. Some tools have predefined risk categories as not all identified risks should be treated the same. Some identified risks are more likely to occur, and some, if realized, would have a bigger impact. Risk analysis and management depends on the types of risks being considered.

Within the context of the technological and business perspectives, there can be distinguished different categories of software risk, for example:

- *Technical risks* that associated with the performance of the software product and include problems with languages, project size, project functionality, platforms, methods, quality, reliability and timeliness issues. Even if there are no mid-project changes in scope, unforeseen technical complications can also turn the project upside down. Project managers might know the technologies they are using in the project very well but still surprises are possible – this component has always been working fine but now when you integrate it with another component, it's a complete mess. The more experienced the technical people are, the lower the risk of unforeseen technical limitations is, but still this risk is always present [10].
- *Standards*, or processes risks may result from excessive constraints, lack of experience, lack of management experience and training, communication problems, organizational issues, lack of authority, and control problems.
- *Financial risks* include cash flow, capital and budgetary issues, and return on investment constraints. These risks are associated with the cost of the software product during software development, including its final delivery, which includes the following issues: budget, nonrecurring costs, recurring costs, fixed costs, variable costs, profit/loss margin, and realism.
- *Personnel risks* include staffing lags, experience and training problems, ethical and moral issues, staff conflicts, and productivity issues. Other resource risks include unavailability or late delivery of equipment & supplies, inadequate tools, inadequate

facilities, distributed locations, unavailability of computer resources, and slow response times.

- *Schedule and scope risks* are associated with the schedule and scope of the software product during development. Changes in scope are frequent in IT projects and to some extent they are quite logical – no matter how detailed your specification is, there are always suggestions that come after you have started the implementation.

Often these suggestions demand radical changes and require change requests that can turn any schedule upside down. In order to address the holistic view of risks, software manager should view the risks from a different viewpoint and then get complete information. Also the scope can be affected by technical complications. If a given functionality can't be implemented because it is technically impossible, the easiest solution is to skip this functionality but when other components depend on it, doing this isn't wise.

Risk identification and risk assessment should be done as early as possible to minimize negative deviations and to maximize positive results during project development. Assessing software risks means determining the effects of potential risks. For the purposes of risk assessment the automated tool might provide predefined set of criteria that would help the experts to conduct evaluation. Risks should be assessed by two dimensions - probability and impact. The project team will take these two dimensions and multiply them together to generate a risk score, so the risks can easily be ranked and ordered, allowing for the team and sponsors to dialog about how to respond to each risk. The Risk Score helps us determine a sense of priority amongst the risks. If, for example, the first risk has a score of $100K and the second of $160K, then the second risk represents a bigger threat to the project's baselines and has bigger priority.

For each risk assessment, the project team must establish how the actual assessment (root cause identification and risk analysis) will be conducted. Having teams outside the project team may be appropriate if the resources needed to do the assessment are beyond those available from within the program team. This team is the core group of individuals who will conduct the risk assessment and normally includes individuals with expertise in systems engineering, logistics, manufacturing, testing, schedule analysis, and cost estimating.

The most widely used technique, that supported by almost all risk management tools is called "Risk map" or "Risk severity matrix" that assess risk probability / likelihood and impact of the potential risk (fig. 2).

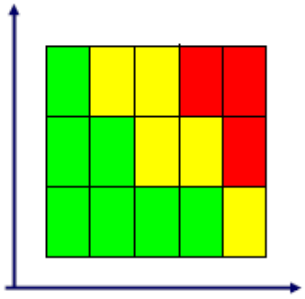

Fig. 2. Risk severity matrix

Red zone identifies the most important events, yellow zone lists risks that are moderately important and green zone events probably can be safely ignored. Project manager might tune the automated tool to customize which combinations of probability and impact result in a risk's being classified as high risk (red), moderate risk (yellow), and low risk (green).

For the extended quantitative analysis you might need more complex tool functionality, such as:

- Monte Carlo simulation;
- Sensitivity and Scenario analysis;
- Sensitivity analysis;
- Probabilistic branching;
- Tornado charts and scatter plots;
- Conditional "If-Then-Else" capability;
- Risk histograms;
- Six sigma functions, etc.

The team members assess (identify and analyze) risks and their root causes using documented risk assessment criteria. An ongoing/continual risk assessment is highly recommended, and is useful during all phases of a program's life cycle. A tailored program risk assessment should be conducted for each of the applicable technical reviews and for each key program decision point. Also project team members report and recommend appropriate risk mitigation strategies for each identified root cause, and estimate funding requirements to implement risk mitigation plans with further documentation and knowledge sharing.

The activity of mitigating and avoiding software risks is based on information gained from the previous activities of identifying, planning, and assessing risks. Usually the predefined mitigation action categories, such as avoidance, minimization, transference, limitation, etc. are available. The types of responses can vary depending on the chosen methodology, but the main four types of responses are:

1. Mitigate the Risk – incorporate specific plans into the project scope to deal with the occurrence of, or to minimize the likelihood of, the risk occurring;
2. Avoid the Risk – remove scope that includes risk from the project;
3. Share the Risk – transfer ownership of scope to another party so they now have risk;
4. Accept the Risk – do nothing, run the chance of the risk occurring, deal with it if it does.

Risk reporting is based on information obtained from the previous topics and compares risk status against previously identified risks. Risk reporting provides capabilities to visualize risk information in graphs and charts that can be further exported to Excel, Word, and PowerPoint in native chart format for easy distribution to others.

Risk monitoring and documentation of lessons learned finalize the risk management processes. Usage of the risk database from past projects to plan current projects can help the managers to avoid most already known problems and lets them learn not from their own mistakes, but employ best practice experience and project expertise.

The final phase, improve and expand, starts when basic risk management practices have been implemented in the project. Improvement is needed to ensure that risk

management is more and more integrated into normal project risk management and to make the processes, methods and tools more effective. Lessons learned should be documented. Also continuous training and facilitation is required. Risk management should also be expanded to other projects within the organization.

Effective risk management requires involvement of the entire program team and may also require help from outside experts knowledgeable in critical risk areas such as threat, technology, design, manufacturing, logistics, schedule, cost, etc. Overall the extended project team carries out risk management and mitigation activities. Risk management is the responsibility of the Project Manager. However, all project stakeholders should participate in the risk identification and analysis process.

External experts may include representatives from the user, laboratory, contract management, specialty engineering, test and evaluation, logistics and industry. End product users, being essential participants in program trade analyses, should be part of the assessment process so that an acceptable balance among performance, schedule, cost, and risk can be reached. A close relationship between the project team and industry, and later with the selected contractor(s), promotes an understanding of program risks and assists in developing and executing the management efforts.

1.2 How to select an automated risk management tool

If you have more than one surname, please make sure that the Volume Editor knows how you are to be listed in the author index.

In order to offer high-quality software products to the market on time and as per the market's requirements, it is important to find computer-based tools with high accuracy probability to help managers make their decision. Software risk analysis and management can be partially transferred into data analysis or data mining. Automated tools are designed to assist project managers in planning and setting up projects, assigning resources to tasks, tracking progress, managing budgets, requirements, changes and risks as well as analyzing workloads.

What should be the selection criteria for an automated tool? It depends on the purpose of risk management in the given software development project and the needs of the team members.

Risk analysis and management are usually based on the information collected from traditional knowledge, or similar well-known cases, common sense, results of experiments or tests, reviewing of inadvertent exposure. The first thing for the automated tools is to collect historical data to build up a database. Once the database exists, it will process the data and mine some useful information to help the manager analyze risks and make decisions. Today's tools can automatically store all project results in a central repository shared by all users. Requirements and changes can be edited, specified and prioritized. Tasks are derived from requirements, which can be traceable through the entire life cycle. This means that *data storage and analysis* should be an important criterion when choosing the system.

Specified risk management software sometimes contains features for test management and quality assurance of the project. Special views and individual reports help project managers assign resources to tasks even in a multi-project environment. Integration with existing testing, quality, cost, and schedule applications

might be essential for identification of related testing, quality, cost, and schedule risks. In this case we need to check if the tool has in*tegration or compatibility with specific applications*.

Most of risk management software supports core risk methodologies, such as CMMI, SPICE, PRINCE2, COBiT etc. For example, the Software Tools Complex for Evaluation of Information Systems Operation Quality (CEISOQ) is designed in accordance with the ISO/IEC 15288 standard and is used to evaluate probabilities of "success", cost, time and quality risks and related profitability and expenses. *Supporting guidance and standards* might be the other important criteria for system selection.

Supporting guidance, standards. and risk methodologies would help users solve on the scientific basis the following practical issues in the system life cycle: analysis of quality management systems for enterprises, substantiation of quantitative system requirements to hardware, software, users, staff, technologies; requirements analysis, the evaluation of project engineering decisions; investigation of problems concerning potential threats to system operation including information security and protection against terrorists; evaluation of system operation quality, substantiation of recommendations for rational system use and optimization etc. [1].

There are lots of methods in Machine Learning study. For example, clustering skills are used to assign risk label to different risks. In each cluster, risks may have similar attributes. Association rule method is used to analyze each cluster to find the relationship between risks and risks factors. Some other artificial intelligence methods (9K-near neighbor approach, ID3 decision tree, Neuro-Network, etc) are used to build risk assessment models and to predict risks of software development. *Supporting specific functionality* might also be a criterion when choosing a tool.

Sometimes, in environments where risk assessments are performed but are not standardized, risk evaluations may vary from one assessor to the next. Whether an appropriate action is taken depends on the particular assessor, meaning that similar issues may end up being treated differently. To avoid inconsistent risk assessments a single system should be used to collect and manage risk management related activities. The system should guarantee that corporate risk tolerance thresholds are employed and followed for risk-related activities across the whole IT project. Thus, we can define important criteria of *standardized risk calculation tools and methodologies*.

In the market, there are all kinds of popular software for decision making that is also applicable for risk management in software risks analysis even if with certain limitation. Microsoft SQL Sever, Crystal Decisions, Microsoft OLAP/Analysis Services are successful decision making software used in business domain. As software development risks can be viewed from different perspectives, even though different from banking, trading business, it does a kind of a hybrid business.

Different criteria may be used while selecting an appropriate risk management tools, such as:

- Data storage / centralized repository and data analysis engines?
- Check lists for software risks?
- Risk assessment capabilities?
- Integration points? (e.g. Microsoft Project / Excel integration)
- Use of graphs and charts?

- Monte-Carlo simulation (other critical functionality)?
- Customizable features?
- Web-based?
- Compatibility?
- Reporting techniques?
- Supporting guidance and standards, etc.

Among specialized risk management software the most popular are: Risk+, Risk Radar®, Risk Watch, IBM Rational Portfolio Manager, OCTAVE-S, CRAMM, Citicus ONE, SCIENTECH, @Risk, ClearRisk, Primavera Risk Analysis, Active Risk Manager, withRISK, Protecht.ERM, Risk Wizard, etc.

There might arise on obvious question: does it make sense to buy off-the-shelf, packaged applications or it is cheaper to create custom solutions (built in-house or with outsourced developers)? To make the decision, several factors should be considered between build and buy. We should analyze technical capabilities, time to market, functionality, support ability, conduct financial analysis and calculate return on investment. Of course, packaged software has many benefits: it is available for many common information technology needs, tested and proved. Most of the packages today come with the global best practices, future upgrades and support, but they not always show a perfect fit with business needs. That's why it is also reasonable to understand the degree of compliance of tools with traditional business processes and rules.

1.3 Tools evolution

Today we have a great choice of different technologies and may use software as we need. Many software users prefer computer tools with much lower setup time. They want to forget about installation, implementation, training and maintenance efforts.

Today, the value is not defined as much by functionality anymore but by connectivity. The user seems to move from process focus and client server architecture to distributed functions and data centric software with real-time connectivity. As business applications continue to mature, a number of new technology and technology adoption discontinuities (social, cloud and mobile) are providing opportunities for users and vendors alike.

While interest in cloud technology and cloud economics abounds, Forrester [11] believes that cloud computing's greatest benefits will come from changes to the IT technology and organizational model. IT decision makers expect the external cloud to play a major role in hosting selected application workloads in the near future. Having the choice between enterprise solutions and web solutions, risk managers might decide on appropriate functional option and price.

As IT project is usually a temporary initiative with defined beginning and end dates, sometimes it makes sense not to purchase a standalone automated software tool for the purpose of risk management, but to get an instant web access to all required functionality, such as identifying, analyzing, tracking, mitigating, and controlling project risks. Choosing web-based solution customers do not pay for owning the software itself but rather for using it.

Where customers may have little interest or capability in software deployment, but do have substantial computing needs, hosting services (SaaS – Software as a Service) is an attractive option. Activities that are managed from central locations rather than at each customer's site, enabling team members to access applications remotely via the *Web*. It allows risk management team to concentrate on their day-to-day software development activities, rather then conducting risk management tool support, administration, security monitoring, new techniques implementation and training.

Web application service hosting allows decreasing large upfront costs as it usually provides free trials, also contains no install costs (only one-time costs) and includes operating costs only.

The lesson of cloud computing is that relatively cheap hardware using virtualization and embedded functionality brings flexibility in computing, storage and network capacity, and management, and enormous improvements in administration.

The main advantages of web-based risk management tools are quick and easy installation and setup and intuitive user-oriented design, when the user can start entering and managing the risks in a very little time. System access via a secure internet/intranet connection makes deployment quick, convenient and easy.

The functionality of such systems usually supports the main risk operations, such as categorizing, prioritizing, modeling, tracking and reporting identified risks. Global availability, access to the software from any machine, cost saving (as there are no hardware costs) and lack of IT support make web-serviced risk management solutions more and more popular. Of course, such services have disadvantages, such as low customization possibilities, work with predefined functionality where managing and tracking functional changes is a challenge, difficulties with other applications integration. This explains the fact, that many risk management software providers have a range of software solutions – from "heavy" client/server application with risk check lists and extensive range of risk management capabilities currently available to "light" web solutions that enable easier and cheaper access to the core risk management functionality. This does not mean that a single web-based tool for managing risks and opportunities cannot be used to meet all the needs of different IT stakeholders and team members. It depends on the risk management purpose and the scale of risk management activities and parties involved, so a customer may choose exactly the solution he or she needs.

Software development projects can be compared to small enterprises, which are normally early adopters of hosting-services solutions because:

- They can't afford to purchase the costly in-house developed / packaged solutions;
- They do not have time and effort for support and maintenance an auxiliary IT solution;
- They enjoy easy web-access and access from mobile devices at any time;
- The cloud technology is good enough to meet the needs of a small team.

After an IT development project is over and there is no need to manage IT risks, the IT project team can easily unsubscribe from the risk management services after paying all rental license fees for the application to the service provider.

Summary

Today we have a great choice of different technologies and may use software as we need. Many software users prefer computer tools with much lower setup time. They want to forget about installation, implementation, training and maintenance efforts.

It is a common opinion that an IT project is always over budget, behind schedule and unreliable. Usage of additional automated tools for risk management purposes sometimes seems useless and costly and project teams face the possibility of losing critical risks, poor communication of risks issues between the interested parties, lack of interest from senior executives due to inability of proper analysis and reporting. This happens because software development and implementation is a complicated process which involves many concerned parties with different expectations. A typical IT project has many interdependent components and modifications, and delays in one component can easily affect everything else. An automated risk management tool does not guarantee success, but serves the primary goal of storage and analysis of identified issues, timely responding with sufficient lead time to avoid crises, involvement of all interested parties into the risk management process, so that it becomes possible to carry out a project that meets its target and provide users and managers with greater confidence in IT. The proposed risk management tools and methods help project managers deal with risk management programs in a most effective and efficient manner.

It is clear that each software project is unique and needs adaptation and customization of selected automated tools to its practical implementation. In this article the author recommends to define a set of selection criteria for automated tools. This set of criteria should be defined for each specific IT project and consider the goals of implementation, objectives of risk management process, the size of IT team and their needs, the possibility of integration with required standards and methodologies, necessity of web access, integration possibilities with Office applications, etc.

Also we should consider the growing demand for service hosting solutions and web-applications which are licensed for use as a service and provided to customers on demand. Cloud technologies become increasingly popular in software deployment as companies prefer to run software on a vendor's or service provider's server with payment based on subscription or time used instead of an individual license. Using web services users interact with the software via a portal on their laptops or mobile devices, almost any software including risk management tools can be run using this method and show good results.

References

1. Avdoshin S., Pesotskaya E., Business informatization. Managing risks, Moscow: DMK Press, 2011, 176 p. [in Russian].
2. Chapman C.B., Ward, S.C. Project Risk Management, Processes, Techniques and Insights, 2nd Edition. John Wiley. Chichester, UK. 2003.
3. Conrow E.H. Effective Risk Management: Some Keys to Success, 2nd Edition. American Institute of Aeronautics and Astronautics. Reston, USA. 2003.

4. Cooper D. Tutorial Notes: The Australian and New Zealand Standard on Risk Management (AS/NZS 460). Retrieved: may 2004 from http://www.broadleaf.com.
5. Del Cano A., De La Cruz M.P. Integrated methodology for project risk management. Journal of Construction Engineering and Management. 2002, 128(6): 473-485.
6. Haimes Y.Y., Kaplan S., Lambert J.H. Risk filtering, ranking and management framework using hierarchical holographic modeling. Risk Analysis. 2002, 22(2): 381-395.
7. Kwak Y.A. Stoddard J. Project risk management: lessons learned from software development environment. Technovation, 2003, 24: 915-920.
8. Pipattanapiwong, J. Development of Multi-party Risk and Uncertainty Management Process for An Infrastructure Project. PhD Thesis, Kochi University of Technology. Kochi, Japan. 2004.
9. PMI (Project Management Institute). A Guide to the Project Management Body of Knowledge (PMBoK). Newtown Square. Pennsylvania, USA. 2004.
10. Xiaomeng Lian, Software Project Management – Risk Management (Abstraction), http://www.docstoc.com/docs/24840578/Software-Project-Management.
11. Jean-Pierre Garbani, Marc Cecere, Forrester, May 3, 2011, IT Infrastructure And Operations: The Next Five Years.