VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SOFTWARE PROJECTS RISK MANAGEMENT SUPPORT TOOL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE   BC. ONDREJ GABRIŠ
AUTHOR

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁSTROJ PRO PODPORU ŘÍZENÍ PROJEKTOVÝCH RIZIK
SOFTWARE PROJECTS RISK MANAGEMENT SUPPORT TOOL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE   BC. ONDREJ GABRIŠ
AUTHOR

VEDOUCÍ PRÁCE   doc. RNDr. JITKA KRESLÍKOVÁ
SUPERVISOR

BRNO 2011

Abstrakt

Management projektů a jejich rizik je v současnosti rozvíjející se disciplína, která si získává stále větší pozornost a uplatnění v praxi. Tato práce popisuje úvod do problematiky řízení rizik, zkoumání metod jejich identifikace, vyhodnocení a managementu, předcházení jejich následkům a jejich zvládání. V další části práce byla provedena analýza vzorků rizik z reálných projektů, byly popsány metody pro identifikaci a vyhodnocení následků rizik v úvodních fázích softwarového projektu, taktéž byly popsány atributy rizik a navržen způsob jejich dokumentace. V závěrečné části zadání byl navržen a implementován prototyp modelové aplikace pro podporu managementu rizik softwarových projektů.

Abstract

Management of projects and contained risks is a fast developing discipline, which attracts increasingly broader attention and gets more prevalent practical implementation. This work describes introduction to the problem of risk management, examination of risk identification methods, risk evaluation and management, preventing risk consequences and their handling. Following part contains analysis of real projects risk sample, description of methods for risk identification and evaluation of risk consequences in preliminary phases of software project. In subsequent part, attributes of these risks were depicted and suggested a way of risks documentation. A prototype of application tool supporting software project risk management was designed and realized in the last part of this work.

Klíčová slova

Projekt, riziko, pravděpodobnost, následky, projektový management, management rizik, analýza, identifikace rizik, vyhodnocení rizik, atributy rizika, matice skóre, dokumentace, opětovné ohodnocení, Metoda kritického řetězce, Teorie omezení, aplikace, návrh, požadavky, případ užití, prototyp, Microsoft SQL Server, Microsoft SharePoint, sdílení, migrace, reálné nasazení.

Keywords

Project, risk, probability, consequences, project management, risk management, analysis, risk identification, risk evaluation, risk attributes, scoring matrix, documentation, re-evaluation, Critical Chain Method, Theory of Constraints, application, design, requirements, use case, prototype, Microsoft SQL Server, Microsoft SharePoint, sharing, migration, real production service.

Citace

Software Projects Risk Management Support Tool

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením doc. RNDr. Jitky Kreslíkové.
Další informace mi poskytl Ing. Zdeněk Fiedler, společnost Allium, s.r.o.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

………………………

Bc. Ondrej Gabriš

10. 1. 2011

Poděkování

Rád bych v této části poděkoval vedoucí mé práce doc. Kreslíkové za její rady a odbornou pomoc.
Dále chci poděkovat Ing. Fiedlerovi z Allium, s.r.o. za poskytnuté konzultace, materiály a neocenitelné praktické zkušenosti.

Contents

1   Preface

Nowadays, all great creations in industry, manufacturing, science, research, development or social sphere are realized using the paradigm of projects.

Project can be described as an extensive effort carried out by more people in order to reach a designated goal. It is a process of using allocated resources and transforming the inputs into an output, the project goal. This process can be rather complicated, especially bigger or long-lasting projects can become very knotty. That's why these complex processes have to be managed, to introduce a system and a way of governance into a wide range of activities and factors influencing the course and outcome of the project. The same applies to IT projects as well. They must be managed effectively and responsibly with prudence, to reach the project goal and deliver the product on time and within the costs allocated. Due to the fact that there is mostly a substantial amount of money invested in project, it is highly coveted to bring the project to a successful end.

But the projects are endangered by risks, being true also for IT projects. This thesis examines the influence of risks in IT projects on the project outcome and project flow in time. An overall topic of mastering of project risks will be elaborated. At the beginning, the theoretical base necessary for subsequent parts of the thesis will be introduced and brief overview of project and risk properties will be presented. After risk general attributes and identification description, two progressive concepts of risk management will be characterized: Theory of Constraints and Critical Chain Project Management method. In the following chapter, examples of typical IT project risks from real environment will be stated.

Methods of risk identification, evaluation, classification and mitigation together with dealing with the consequences using the four quadrant graph are described in next chapter. Then a design and implementation of application prototype for risk management support follows. The application should stress the time-related risk management aspect in IT projects. Then, in the end of the thesis, the results will be evaluated and discussed.

This thesis was elaborated as a continuation of the Term project [12] which comprised the first four points of this thesis assignment. The thesis takes this elaboration over in chapters 'Risks in IT projects', 'Critical Chain Project Management', 'Typical project risks' and 'Means of identification and evaluation of risks', extending the work to cover the rest of the given assignment and summarizing the achieved results in the end.

2   Risks in IT projects

2.1   What is a risk?

We are living in world of consumer-information society. Time was when money represented the greatest value, having money was the key to success. The human mentality had this basic orientation – gathering the money meant some kind of warranty to the successful and contented life of human being. The business was all about the same, companies struggled to produce more and more goods or services, sell their products and gather more and more money, no matter what was going on in the around world. They did not need to look around, the money guaranteed the profit and future growth as well.

Nowadays, situation is different. The money is still a key goal, one of the most important in our society and business for sure. However, the process of achieving it has changed. The world out there is tougher, resources are limited, the time is pressing on us. We need to try harder to get closer to the goal, step by step. There is no space for mistakes, competition is waiting for each and every one of them. That's why the money itself is not a key to success any more, it is just not enough. The key factor is information or we can say intelligence. We simply need to know, know sooner than others and be certain about the knowledge. Comprehensive knowledge and precise information is what matters today, besides money. Everything needs to be known, everything exactly planned. Any uncertainty in business can be dangerous in its way. And here we come to the definition of a critical term, a risk. Risk is basically an uncertainty in business ventures [1]. As this thesis is related to software projects, we will consider risks as uncertainties in the software projects and their management. In today's world of business, company has to have precise plans for its strategy and development. This applies doubly to the projects, also those in IT.

Progress, alongside with constant development in performance and management of IT projects brings unknown factors that can influence both the project and its outcome. These factors can occur with a certain probability and we need to consider them in our calculations, as they can have both positive and negative effect on the project or product. Positive effect usually means an improvement in project flow or product quality and in general represents a gain. A negative effect, however, is understood as jeopardy of degrading product quality or smooth project flow. Therefore, its final result for the project owner is a loss. An assessment of the risk then represents the relative probability of loss compared to the probability of gain [1]. This is just a very basic point of view, though. There are various other factors to consider when assessing risks from the wider point of view. These can include the company's goals, gains vs. relative losses or opportunities expected. To extend the point of view even more, we can consider factors like magnitude of loss compared to the expected gain, company's tolerance to losses, contribution of undertaking a risk to the company's goals and many others. Then the risk can be defined as the combination of probability of occurrence and the magnitude of loss it causes [1]. This is also called a risk exposure. Assessing the risk is always connected with a certain, higher or lower, rate of indeterminateness. That's because an occurrence of the risk can only be framed in a certain range of expectancy and risks in general occur for the sake of circumstances that are hardly predictable or beyond. When a risk strikes, it threatens a process (or more processes),

whose outputs can be affected. If the process is essential for the project, then the risk is jeopardizing also the outcome or performance of the project itself.

When it comes for the responsibility for risks, each and every risk that is to be managed must have an owner. The owner of the risk is responsible for managing and treating the risk properly. It is usually a person (manager) owning a process that is threatened by the risk. Risk owners are not supposed to run away from the risks discovered, they rather take the advantage of being aware and utilize the positive aspects of it.

2.2 Why are most IT projects finished late?

Statistic results of IT projects are surprisingly poor. IT is generally considered a young, very fast developing area, using latest technology and progressive approach reflecting the latest technological, scientific or management findings. The hardware is more and more powerful, IT researchers are pushing physical boundaries when developing next generation chips. Software development is not falling behind, one can find or have programmed software for anything, according to his needs. But what about IT project management itself? IT projects drawn in problems. Only a minor part is considered successful at the end. Vast majority of all IT projects is finished late, over-budget, temporarily stopped or completely cancelled before finishing. Yet, we have the best strategies worked out, the best managers hired and project teams members carefully chosen. Everything is planned, time and money reserves added, contracts signed and projects are supposed to run smooth and end well.

Obviously, something is wrong. We are missing something important. When we look into the details of the failed or unsuccessful projects, we can see the project analysis was in general done correctly, time and cost estimates were reasonable and project team was well assembled. But something wrong happens in most cases, what was not included in the preliminary calculations. These complications cause either cost increase or time delay in the project. Or both, if the experienced issues are serious in quantity or quality. And these unexpected issues are what we call risks. They cannot be predicted, they are beyond of what we can foresee and some of them cannot be prevented at all, as we are unable to influence their causes in any way. Because of them, projects do not finish in scheduled time and within estimated cost range, even though seemingly sufficient reserve funds were allocated at the beginning. Therefore, the project risks have to be managed in order to bring projects to a successful end. Project risk management is a way of managing project with respect to the risks that can occur, their probability, severity, consequences and methods of their preventing and dealing with potential consequences. There are more options how to cope with risks in project management, they vary in attitude and successfulness.

2.3 Risk management

In every performed project, there are multiple risks that can occur, each having its own probability and severity of consequences. These risks, if they ever occur, need to be handled properly in order to eliminate their repercussions within the running project. And that is the exact purpose of the risk management process as the definition says: "Risk management is a systematic approach to reducing

the harm due to risks, making the project less vulnerable and the product more robust" [1]. In another words, risk management is working on the task of preparing the project itself, people involved and procedures planned for the occurrence of potential risks and dealing with them in a way that causes the least possible damage to project goals. The risk management, however, is not trying to eliminate all possible risks at all costs. As a matter of a fact, some cannot be eliminated to zero probability at all. Some other's occurrence chances can be reduced to a reasonable value, some can be prevented entirely. Whatever the case is, risk management tries to determine ways how to handle the risks and minimize the losses it causes. Another definition of risk management defines it as follows: "The purpose of project risk management is to obtain better project outcomes, in terms of schedule, cost and operations performance" [2].

But there are two sides of the same coin. Each and every unmanaged risk represents a weak spot in the system that needs to be fixed. From this point of view, each risk can be understood as an opportunity to improve the process. Risk management is there to highlight the sources of uncertainty in the project and make them visible and understood to everybody it may concern. When working correctly, the risks should be identified in time, taken care of and the responsible persons should be able to make informed decisions while dealing with them [6]. This way, every identified risk that is not treated yet, represents an empty space in project management, which needs to be filled and respective risk dealt according to the strategy prepared. After each managed risk, system becomes more risk-proof and therefore more robust, which contributes to a better project outcome.

The risk management process starts at the very beginning of each IT project. It should be a consistent, pervasive effort integrated with all activities within the project phases. Also particular activities throughout the project should be reviewed repeatedly from different angles and levels of detail.

2.4 Identification of risks

The process of identifying a risk consists of two basic parts: first one is searching for the risk anywhere in the project [1], where it could appear. Second part includes identification itself, defining the risk, clarification of its properties, which will help with classification. Searching for risks might not be a simple task. Quite often they can be hidden or evading recognition, sometimes they don't fit the expected shape or nature and can be overlooked. Sometimes their effects or symptoms are not visible and therefore the risks are not noticed and identified. That's why we need to consider all angles of view, various levels of details, all options and all directions from which the risk could come. After a risk is found, we need to clarify its attributes, classify it into system and rank the risk according the quality and quantity of the consequences. After the risks identification is carried out, we receive a validated risks list. Prioritizing of the risks in the list should be done with a reference to the additional information obtained by a brief preliminary analysis of particular items.

2.4.1 Risk identification methods

Methods for identification of risks in specific projects can be divided into two general types [1]. First type of risk identification methods is broad and generic, it involves risk search in both internal and

external environment of the project. The scope of this search exceeds the boundaries of current project and goes further away through entire business. The second type, on the contrary, works within tightly defined context and represents a formal method divided into 6 phases.

Type One

This type of identifying risks is open-minded and generic [1], using intuitive as well as historical methods. It covers both internal and external risks. The aim of this type of search is to uncover the risks unknown before the analysis.

Intuitive methods:

- Mind mapping

- Interviews and focus group discussions

- Brainstorming

- Out-of-the-box thinking

- Analogy

History-based methods:

- Top ten risks

- Risk checklists

- Taxonomy-based questionnaire

- Examining of similar implemented projects experience, reports and audits

- Scenario analysis

- Work Breakdown Structure analysis

Type Two

The second type of approach is more formal, structured and specific. The results are retrieved in six consequent phases [1]:

1. Context setting

2. Data gathering

3. Risk discovery

    a. Mapping

    b. Risk survey

    c. Risk models

    d. Risk intelligence

4. Attributes assignment

5. Validation

6. List

2.4.2   Risk identification levels

Identification of risks is definitely a process dependent on the depth and range of information taken into account as well as level of abstraction at which we expect the results at.

There are three basic levels of risk identification:

- Process-level risk identification

- Project-level risk identification

- Enterprise-level risk identification

2.5   Risk analysis

Risk analysis is a process referring to identified project risks in context of an IAMT (Identify, Analyze, Mitigate, Track) cycle. Risk analysis starts when project risks have been identified and their attributes set. In its first stage, the purpose of risk analysis is to understand risks better and to verify and correct risk attributes [1].

Analysis of the risk is crucial to the next step in IAMT cycle – mitigation. So the purpose of risk analysis in its second stage is to select a risk for being mitigated. This is done by assigning each risk a priority rating, in view of running processes, ongoing activities or plans established to reduce or control the risk. Severity of a risk can be determined as a combination of its consequences or impacts on project objectives, and the likelihood of those consequences arising [3]. This can be accomplished using various techniques ranging from simple qualitative approaches to more sophisticated quantitative ones.

Basic overview of risk analysis types [3], starting from the qualitative-only:

- o Qualitative analysis is based on nominal or descriptive scales for describing the likelihoods and consequences of risks. This is particularly useful for an initial review or screening or when a quick assessment is required.

- o Semi-quantitative analysis extends the qualitative analysis process by allocating numerical values to the descriptive scales. The numbers are then used to derive quantitative risk factors.

- o Quantitative analysis uses numerical ratio scales for likelihoods and consequences, rather than descriptive scales.

2.6   Risk evaluation

Risk evaluation is a process of deciding, whether the project risks can be tolerated or not, with consideration to the related factors affecting it. When evaluating a risk there are these factors to bear in mind [3]:

- the controls already in place or included in project plans

- the likely effectiveness of those controls

- the cost impact of managing the risks or leaving them untreated

- benefits and opportunities presented by the risks

- the risks borne by other stakeholders

The evaluation in its basics draws a comparison between risk priorities from initial analysis and all the rest of risks, organisation's priorities and requirements. The product of this comparison is a concurred list of risk priority rankings. Risk priority defined in initial ranking can be increased, if the risk is found more important than the preliminary ranking indicated or if it is related to some highly ranked risks and needs to be treated jointly with these ones. Another reason to increase the ranking may be to make some risks more visible if they need some sort of special handling. Decreasing the risk ranking can be mostly the case of those ones, which are frequent and strongly awaited to strike, but have only few unfavourable consequences and standardized action plans for dealing them are defined.

Team working on the project has the best knowledge about risks related to it. After going through the analysis and evaluation there should be no errors or inaccuracy left in the risks priority rating. Risks are then to be treated according to their priorities and each of them should have a manager for supervising examination and evolvement of treating possibilities.

3   Critical Chain Project Management

Critical Chain Project Management (CCPM) is a method of project planning and management developed by Eliyahu M. Goldratt and described in his book named Critical Chain in 1997. Compared with more traditional methods of PERT or Critical Path, CCPM pays maximum attention to the resources needed to fulfil the project tasks and doesn't insist on strict scheduling and task order [4]. The method maintains the project resources levelly loaded, requires them to shift quickly between tasks and asks for flexible start times while working on tasks. The Critical Chain method is basically application of algorithms and methods introduced in Theory of Constraints (TOC) to the project management. When compared to the traditional methods of project management, correctly applied CCPM method achieves far better results, vast majority of projects is finished on-time and on-budget. Moreover, CCPM reduces effects like Student's syndrome, inappropriate multitasking, poorly synchronized integration, lack of prioritization, In-box delays, represses Murphy's Law, Parkinson's Law or redundant resource allocation.

3.1   Theory of Constraints (TOC)

Theory of Constraints is a management philosophy, which was introduced by Eliyahu M. Goldratt in 1984 in his book The Goal. As obvious from the name, this theory comes with management of constraints which are present in any system working to achieve some kind of a goal. According to the theory, there are a small number of constraints in each system, at least one. On the other hand, Goldratt claims we would have extreme difficulties to find a real working system with more than three constraints. The TOC is trying to identify the constraint, subordinate whole system to it and break the constraint, all in five iterative steps. One of the new key points in TOC method is the critical path – the longest path needed to complete the project. To create it TOC is utilized by adding together the predecessor/successor constraints (task durations). Another difference brought by TOC is the change in measurement of companies based on measures as throughput, operating expense and investment.

3.1.1   The five focusing steps of TOC

The TOC says that there is at least one constraint in every working system, preventing it from achieving its goal at a higher rate. The constraint is the "bottleneck" of whole system limiting its performance. The only way to increase performance of the system is to elevate the capacity of this constraint and to increase a total flow. As soon as the prerequisite of defining the main purpose of the system (its goal) and a way of its measurement are fulfilled, five focusing steps can be performed iteratively:

1   Identify the system's constraint – focus on task/resource/process that keeps the organization from achieving the objective in higher rate

2   Exploit the system's constraint – elevate the capacity of the constraint, get as much as possible out of it

3   Subordinate everything else to the above decision – adjust the rest of the system to work according to the decision above

4   Elevate the system's constraint – take other necessary steps to eliminate the constraint

5   If in the previous steps the constraint is broken, return to Step 1. Don't let inertia to cause a system's constraint.

These five steps are concentrated on ongoing improvement of the system's constraints. This process is known as the Process Of Ongoing Improvement (POOGI).

3.2   How does CCPM work?

Critical chain can be defined as a sequence of precedence-dependent and resource-dependent terminal elements preventing the project being completed in a shorter time with finite resources. Creating critical chain is very similar to creating critical path. However, there are some differences between them [4]. To begin with, critical chain considers resource dependencies, while critical path doesn't. Next difference is missing search for an optimum solution. The "good enough" solution is sufficient, because there is no analytical method to find an absolute optimum. Also, the fundamental uncertainty in estimations is incomparably greater than the difference between optimum and near-optimum solution. Another important difference between critical chain and critical path is inserting various types of buffers (project buffer, feeding buffer and resource buffer). One of the most significant differences is measuring project progress by monitoring the consumed part of the buffers and absence of pursuing individual task completion and due dates. This is affordable, as CCPM aggregates a large amount of time reserves into the buffers to secure the due-date performance and save these time reserves from being pointlessly wasted within individual tasks.

Planning a project using CCPM is the same as with critical path method. Tasks are planned and for each task there is a particular amount of time added to the bumpers. Each task is then assigned necessary resources and the plan is resource-levelled. The longest sequence of resource-levelled tasks leading from beginning to an end of the project is identified as the critical chain. Project buffer is placed to the end of sequence, feeding bumpers to the ends of non-critical chain tasks, just before their feedings to the critical chain.

3.2.1   Experimental results of pondering typical risks

When managing project using the critical chain method, following risks can stand out:

- insufficient buffer setting, resulting in project delay
- incorrect determination of critical chain, eventually caused by wrong identification of critical path and conflicting resources
- ineffective resources utilization

- counter-productive approach of the project management – finishing project on time at the cost of minor increase in project costs can be far better than finishing it months delayed but with originally planned expenses. (This is a question of minimising losses after we find out that either the time or budget allocated for the project is insufficient.)

4 Typical project risks

Typical examples of projects risks from several projects are displayed in following tables.

4.1 Project #1

Phase	Risk / Problem	Strategy to eliminate the risk
Analysis	1) Users don't understand (or can't describe) their own processes or requirements	A special tool for creation of diagrams will be used to create a graphic representation of business processes.
Analysis	2) The key users are not skilled enough professionally or they are overburdened by end users requirements or other duties	The Steering Committee will either ensure new key users or arrange more time for the current key users by releasing them from some of their duties. The key users will be motivated to engage in project by special method of financing of the project.
Analysis	3) Users don't agree with the project aim or with the requirements.	It is important to document the current state. Then the analysis of functional requirements must be carried out and signed by the top management.
Analysis	4) The analysis shows that significant changes to some parts of the processes have to be done.	It is important to document the current state. Then the analysis of functional requirements must be carried out and signed by the top management.
Design	5) Incomplete or inaccurate analysis (any reason)	Partial agreements of the appropriate parts will be made during the design phase.
Design	6) The time table was based on incorrect estimations of time consumption and incorrect presumptions.	The Critical Chain method will be used to manage the project and continuous evaluation from the resources consumption viewpoint will be made.
Management	7) The key users do not have enough time to devote to the project or are not available for discussions when needed.	The customer's project manager will set the priorities to the key users that they will be able for a full cooperation on the project.
Management	8) Late or not accepted strategic decision in relation between customer's branches.	Necessity for such a decision will be described in the outputs with a proposal of the deadline. The Steering Committee overviews a decision making procedure.

Table 1. Project #1 risks.

4.2   Project #2

Phase	Risk / Problem	Strategy to eliminate the risk
Analysis	9) Users are not available for interviews and workshops.	Interviews and workshops will be planned sufficiently in advance. Should the problem persist, Steering Committee will deal with it.
Analysis	10) Supplier's analyst does not get along with users.	If the analysis is minor, nothing needs to be done. In case of major analysis the analyst will be replaced by another one.
Design	11) The customer gets wrong ideas of the functionality of the solution and the supplier doesn't mark in the target concept that this functionality is not included in the solution.	Supplier will pay special attention to "Red Flag" areas in analysis. If there are any highly-critical areas, supplier will provide a very detailed description of functionality that customer does and doesn't get in these areas.
Design	12) Incomplete or inaccurate analysis (any reason)	Supplier specifies data extent and deadlines for the data to be available. Project management from the customer's side will assure timely and correct data.
Education	13) Users are not provided sufficient room to master system controls on time and effectively.	Supplier will utilize an examination process and project leader will influence the customer's management, based on the user classification, to increase motivation of users.
Move to alpha-test	14) Functionalities required by key users in previous periods will not be implemented, because supplier didn't have enough time for implementation, they are not described in system design or they weren't communicated.	Regular (approx. on weekly bases) notifications of key users about completed functionality and its transfer into the prototype. Until next notification, key users will be obliged to check and respond to the functionality passed to them.

Table 2. Project #2 risks.

4.3   Project #3

Phase	Risk / Problem	Strategy to eliminate the risk
Realization	15) Testing isn't paid sufficient attention from users and will not uncover defects complicating future system launch.	Key users will be motivated to high-quality testing and acceptance by finishing bonuses derived from number of claims and after-launch problems.
Commissioning	16) Users don't have enough time for checking migrated data and their correction.	Key users will be motivated to high-quality testing and acceptance by finishing bonuses derived from number of claims and after-launch problems.

Table 3. Project #3 risks.

5 Means of identification and evaluation of risks

Risks stated in previous chapter were obtained from three similar projects. Relatively high number of them appeared in more than one out of three example projects. This shows the similarities in all example projects, so there were probably not huge differences between their types. Repeating risks were not taken into the sample more than once. The result we got from the three projects is a set of more or less typical project risks that can be usually found in this kind of projects. In following chapters we will focus on means of identification of these risks and evaluation of their consequences.

5.1 Means of identification

Let us focus now on the methods of risk identification used in example projects. We used informal, generic intuitive and also history-based methods to identify the risks mainly in preliminary phases of projects. These methods included [3]:

- brainstorming
- focus group discussions
- interviews
- mind mapping

From the history-based methods were used:

- similar projects experiences
- risk checklists
- top ten risks

For identification of risks in these examples, using classical means of informal, intuitive methods was suitable. Risks repeating in projects after first occurrence were omitted, only the first appearance of them was kept. The combination of chosen risk identification methods always depends on the type of the project, specific customer, project environment and type of expected output of the project.

For the purpose of risk identification and evaluation, we will merge identified risks from all three example projects into one list. Important for us is the phase of the project stated in each row of the table with risks. It shows the phase of origin for each particular risk, so it is easily trackable. We will use the advantage of tracking risks to the phase of their origin later.

After its identification, every risk should be numbered and briefly described, so that the basic information about it is at disposal without any further necessary examination.

5.2 Risk evaluation

The outcome of risk identification is a numbered list of risks. Next stage after identifying the projects' risks is their evaluation. Risks will be evaluated according to the likelihood of their occurrence and their consequences [6]. Table 4 shows the risk scoring matrix which we will use to evaluate the risks.

Evaluation	Score
High	≥ 18
Significant	12-17
Medium	5-11
Low	1-4

Table 4. Risk scoring matrix.

Scoring matrix	Risk consequences					
Probability of occurrence	Insignificant	Minor	Moderate	Major	Catastrophic	
Almost certain	Medium (5)	Medium (10)	Significant (15)	High (20)	High (25)	5
Likely	Low (4)	Medium (8)	Significant (12)	Significant (16)	High (20)	4
Possible	Low (3)	Medium (6)	Medium (9)	Significant (12)	Significant (15)	3
Unlikely	Low (2)	Low (4)	Medium (6)	Medium (8)	Medium (10)	2
Rare	Low (1)	Low (2)	Low (3)	Low (4)	Medium (5)	1
	1	2	3	4	5	Severity

5.2.1 Evaluation of risks according to the scoring matrix

Following table shows evaluation of the risks introduced earlier:

Risk number	Probability [1-5]	Consequences [1-5]	Risk Evaluation
1	3	5	Significant
2	2	4	Medium
3	2	3	Medium
4	3	4	Significant
5	3	5	Significant
6	2	4	Medium
7	2	2	Low
8	3	4	Significant
9	3	4	Significant
10	1	4	Low
11	2	5	Medium
12	2	4	Medium
13	3	3	Medium
14	2	4	Medium
15	5	4	High
16	4	4	Significant

Table 5. Risks evaluation.

This evaluation of risks is done at the beginning of each project risk analysis. It is useful for getting the picture of possible risks in project, as well as realizing the probabilities and consequences of their occurrence. However, this is far from sufficient for managing the project risks all the way from project start to a successful ending. That is, because this initial analysis is only static. The project, through its life cycle, is acting like a living organism, it is changing and continuously evolving, what brings a wide range of changes also into project risks.

5.2.2   Re-evaluating the risks[1]

Obviously, for effective risk management we are going to need something more complex and sophisticated than only a static risk analysis at the beginning. We must be aware of risks' status, likelihood and possible consequences during whole project duration. It is necessary to re-evaluate the initial risk list several times during the project flow, so that we are aware of the actual situation. The question that arises here is: "What is the proper moment to do the re-evaluation?"

The main reason why re-evaluation of the risks during the project flow is necessary can be illustrated by following example:

A middle-sized company was carrying out a project of critical importance. There were more people included in project team. Initial risk analysis was performed properly and the precautions were taken according to the scores of significant and major risks. There were no critical risks identified, so the project seemed to be rather risk-safe. Project team was performing well thanks to its key member, which was pulling the team together and had a complete know-how of the project activities. Then, because of the external impact, this person had to leave the company. He eventually quitted the job and had to be replaced in project team. The problem was that he took all his developed know-how with him and there was nobody capable of substituting him. Project team could not manage to carry on in the project essential activities and the project started to fail. Finally, the project crashed catastrophically in total failure and it had fatal consequences to the company in the long time horizon.

The problem here was that the risk of project team inability to work was evaluated as "lower than low", in fact. It happened because from the beginning, the project team had an outstanding performance and delivered great results. Thus, the risk of project team operational inability was very improbable. But the risk probability as well as consequences started to rise with upcoming project progress. Probability increased because of external factors, severity of consequences was growing with proceeding project phases. When the key person abandoned the team in the middle of project, both these variables were at maximum. The root cause was that project managers on the customer's side were unwilling to accept this situation can happen and the risk was not re-evaluated, it was ascribed the initial score instead. Increasing impact of this risk was being ignored throughout the project run.

Given example shows a need of dynamic risk analysis. Risks should be re-analysed in a suitable moment, so we know how they evolve throughout the project life cycle. The right moment for re-analysing the risks can be determined by Critical chain project management (CCPM) method, which is a dynamic iterative method, allowing re-evaluating project risks in time. Obviously, risk analysis itself is a complex process and it needs to be done in reasonable time periods. It cannot be done too often, as it would consume majority of time and effort of the project management team. On the other hand, if it is done too rarely, maybe even just once, it may not reveal all the risks threatening the project or

---

[1] Chapter 5.2.2 and subsequent chapters related to risk documentation were based on consultations with Ing. Zdeněk Fiedler from Allium, Ltd.

pinpoint their relevance correspondingly. This shows that determining an appropriate moment for redoing the risk analysis is crucial for the project success.

While managing the project by the CCPM, the proper time for re-analysing the project risks will be found out by tracking the size of buffers. When any of the buffers starts to decrease rapidly, it is an indubitable sign for re-analysing the project risks. Problem reasons can include occurrence of unknown and undocumented risk, changed probability of some of the known risks, increased severity of consequences of these risks or even underestimation of key factors and failings in previous analysis. The great advantage of using Critical chain project management method lies in the fact, that when a buffer starts to be consumed rapidly, it is not already too late. While a certain buffer still exists, its rapid consumption is just an in-time warning that something in project went wrong and it needs to be corrected. Another pattern which should be found during re-analysing project risks is following: risks identified as significant in preliminary phases of the project should be having lower and lower significance in the subsequent phases. In other words, high-scored risks from the initial phases should be already passé in later phases of project. If their score is rising instead of descending, there is a serious problem in the way the project is managed. The weight and severity of consequences of a wrongly evaluated (or for any other reason unmanaged/unrevealed) risk are steeply rising in time and can have fatal effect if the risk occurs. This also contributes to the fact that mistakes in initial phases of projects are of the highest impact.

As an example illustrating such situation can be used a Czech internet mall called Mall.cz, which decided to change their system to SAP. Because of a mistake in initial analysis, the migration of data was not successful and it took 6 months to restore the original data about transactions and purchases. As a result, the company was unable to make out any invoice for more than 6 months and the consequences were severe.

Stated examples show some of the failures in project management that should be avoided by far. Proceeding this way, project risks can be re-assessed in appropriate time and treated accordingly. Each time risk situation of the project changes after re-analysing, newly discovered risks should be countered by taking suitable measures.

When evaluating the project risks, either in initial risk analysis or during re-evaluation in further stages of the project, there is one important factor to consider. The output score of each and every particular risk determined by any method used, including CCPM, is as exact as are the inputs used for computation. Which means the main problem is not to choose the best method, because there is a variety of them and mostly, they are clear, exact and reliable enough. The main problem dwells in determining exact and reliable inputs to be used in chosen methods. Therefore, the main focus must be shifted to determining realistic information about probability and severity of consequences of possible project risks. This must be put the biggest effort into, instead of choosing the best method for risk evaluation itself.

After initial evaluation of the risks and multiple re-evaluations during the project flow we may notice that some risks are highly fluctuating, mostly especially in their probability of occurrence, but possibly in their consequences as well. If this pattern shows up in a project, then there is probably a problem in its risk management or risk countering measures used. That is the case of a number of projects having serious problems caused by ignoring the factor of altering likelihood and severity of project risks. Practise confirms the assumption that high risk volatility itself may become one of the major project risks, endangering the project at most!

5.3 Documenting identified risks

After the risks are identified, they need to be documented in an appropriate way. The key identification attributes of each risk are the likelihood, severity of consequences and the phase of project in which the risk arises. As described previously, we assign each risk a score based on the scoring matrix. Using square scoring matrix with 3 degrees of likelihood of the risks and 3 degrees of consequences severity is generally sufficient for vast majority of projects. We used scoring matrix 5x5 to be more specific and to classify the risks more precisely. Though, the main issue while evaluating risks is not the number of classification degrees, but exactness and reliability of inputs, as described earlier. That's the main reason, why it is no good using scoring matrix based on more than 5 levels of risk likelihood/severity and with a lot more than 5 resultant classes.

5.3.1 Graphical representation

Besides representing evaluated risks and their scores in a table, there is another approach that can be very beneficial, as it offers a different point of view. Graphical representation of risks provides us additional information about each identified risk. Following graph represents positions of risks divided into 4 quadrants, each one with different meaning.

Let's start with the way how risks are identified. Before a proper analysis at the beginning of project, there is just a knowledge about "some" risks that are probable in the project, having "some" (not specified, approximately maybe average) probability of occurrence and "some" severity of consequences (not precisely determined, something in the middle probably). At this point, the risks are very vague and indefinite, nothing specific is known about each of them but the fact they most likely exist. So represented graphically (Graph 1.), it would be a set of risks from *Typical project* risks, somewhere in the middle of the graph.
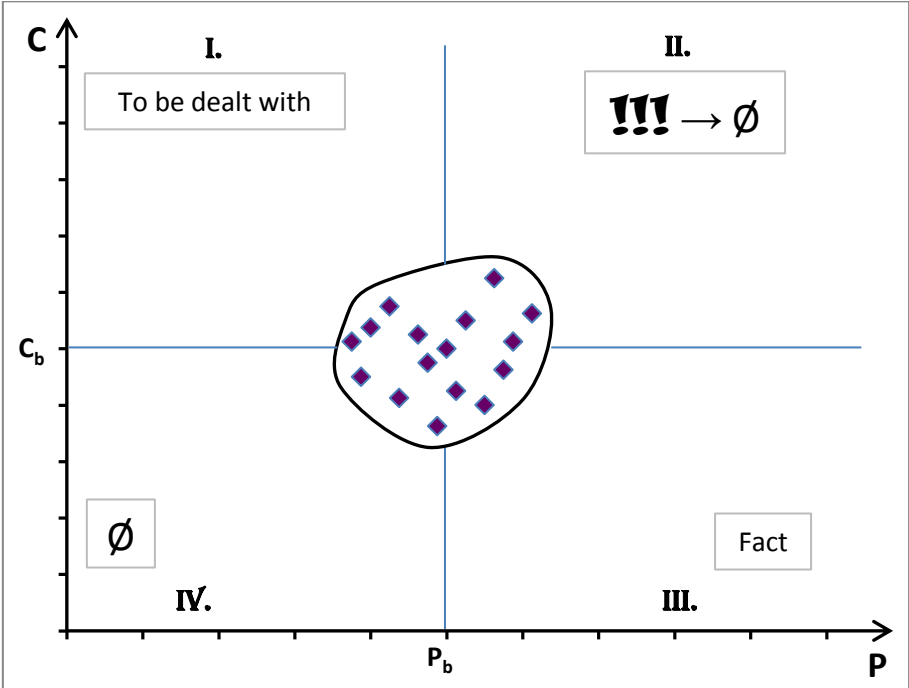
Graph 1. Set of unspecified risks from Chapter 4.

As the risk analysis is proceeding, the risks are described more in detail and assigned some level of likelihood and possible consequences. Represented graphically, that places each risk in a particular quadrant (Graph 2.). The more precise assessment of risk is, the bigger is usually the distance from the former position. In general, minimum of risks should remain directly in the middle of the graph - they should split into particular quadrants. Each of these quadrants has a substantial meaning for future counter-measuring the risks and their possible consequences. The boundary values between intervals are:

- P_b – boundary probability of risks occurrence. Can be set to 50%, but is subject to vary from project to project, depending on project and risk nature, project budget or another factors.
- C_b – boundary cost level, is related to the project budget, tolerated amount of money that can be spent on dealing with handled and unhandled risk consequences. It is also related to the amount of money resources allocated for project and border between the performing company's profit and loss from the project.
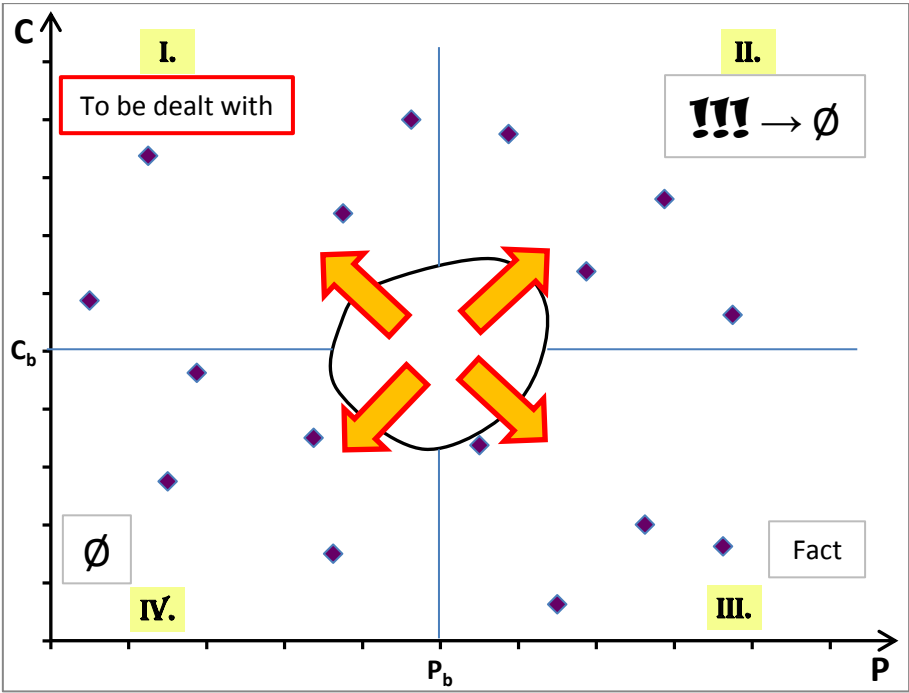
Graph 2. Specified risks divided into quadrants in one of project stages.

5.3.2 Graph quadrants specification

To start with, we analyse **quadrant IV.** Risks belonging here are not of much importance. Probability of their occurrence is low as well as severity of their consequence. If they can be mitigated easily, project team does it. If their mitigation is too pricy or time-consuming, these risks are usually just left alone. In each project there is a reserve budget for cases like this, so if the risk from this quadrant strikes, it will be covered from this budget. Their impact to the project is not too big and this is expressed also by the symbol representing this quadrant in graph – $\emptyset$.

Risks with high probability of occurrence but not severe consequences are in **quadrant III.** Arising of these risks is very likely (almost certain sometimes) and even though the consequences are not too dramatic, there is a need for covering the cases of occurrence by a set of defined counter-measures and rules, so that the risks can be handled. High probability of risks in III.[rd] quadrant implies the name of it – *Fact*. That's because these risks are nearly sure to arise. This is also the reason why money and time expenses required for countering quadrant III risks are directly included in the project budget from the beginning.

Another group of risks is emerging in **quadrant II.** These risks are specific by very high probability of occurrence and very severe, frequently catastrophic consequences. After their identification and evaluation, all of these risks should be mitigated or completely prevented. If their likelihood or consequences (or, possibly, both) cannot be reduced, then re-evaluation of whole project must be done. It is not reasonable to hurl into a project that has plenty of severe risks almost certain in occurrence and fatal in consequences. Such a project has a very small chance to be successful. Thus risks from II.[nd] quadrant should be mitigated and moved to another quadrant or the project should not be launched at all. Either way, after fulfilling these conditions, quadrant II should not endanger the project any more. Therefore, this quadrant is marked "!!! → Ø".

Finally, we come to the risks classified into the **quadrant I.** These are the *risks to be dealt with*, because only this quadrant and risks piercing into it are really endangering the project through its life cycle. These need to be mitigated, counter-measured or dealt with, if they occurred already. So quadrant I risks are the ones that project team must handle throughout the project – quadrant IV risks are too insignificant, quadrant III risks occurrence is included in time and money reserve funds and quadrant II risks are to be eliminated or mitigated rapidly in preliminary project phases (or there is no point in continuing with the project).

Since risks in quadrant II have been eliminated or reliably mitigated, they have probably been moved to quadrant III or I (Graph 3, orange arrows). As stated before, for quadrant III risks there is so-called buffer (both in time and money view) and procedures are defined to deal with these risks effectively or lower their probability to quadrant IV (Graph 3, green arrow). So the project team knows about quadrant I risks and is dealing with them (as effectively as possible). And still, we need re-evaluation. Why is it so? Because the risks are "travelling" through graph quadrants during the project's life.

Movements of risks illustrated by green and orange arrows, especially in preliminary project phases, are intentional and beneficial for the project. Our main concern should be focused on movement along the red arrows, which indicates something in the project went probably wrong or, at least, that the risk situation of the project has changed and something needs to be done. Particularly if we discovered this movement after re-analysing the risks then taking proper counter-measures is in place. Newly arisen risks are severe enough in their consequences, so it is necessary to handle them in an effective manner and lower their impact as well as probability, if possible. If they are unhandled or even unspotted for longer time, they can migrate to the II.[nd] quadrant (without precautions taken, probability will rise) and may cause a disaster with fatal consequences for whole project. But appearance of IV.[th] quadrant (or newly arisen) risks in the I.[st] quadrant is expectable and project team should be able to handle these challenges during project life.
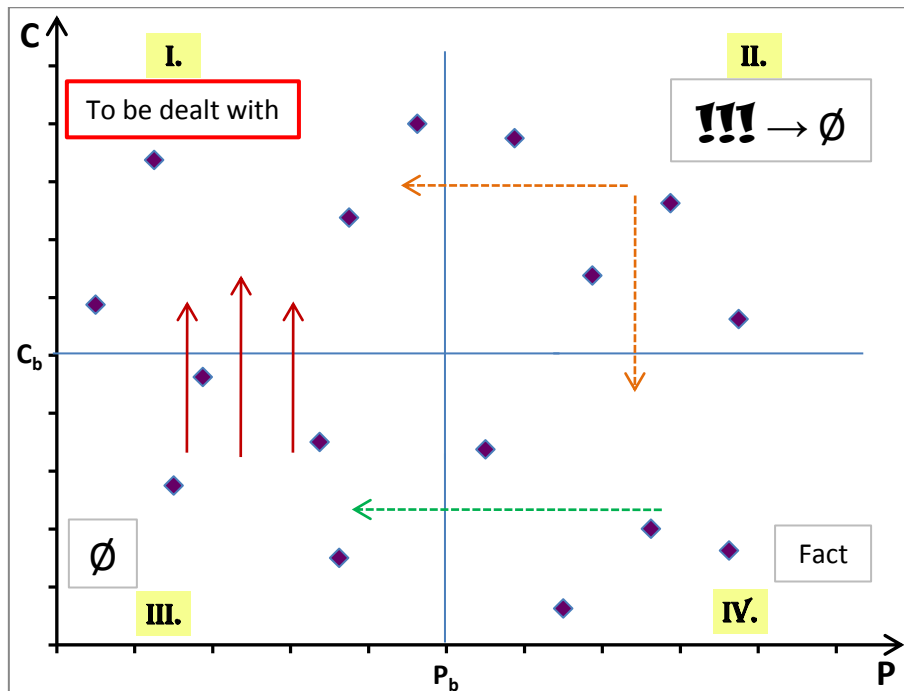
Graph 3. Risk movements through quadrants.

6 Design of application

After defining risk identification, evaluation and means of documentation, this chapter introduces a model of designed application for risk evaluation and management. Going through multiple consultations at Allium, Ltd, the final set of application requirements was derived, based on risk identification, evaluation and documentation approach defined in previous chapters. Paying close attention to analysis of these requirements, an initial, implementation independent model of application functionality was created and represented by use-case diagram.

6.1 Informal specification

Specific requirements for application functionality were obtained during interviews with project leader Ing. Zdeněk Fiedler from Allium company. At the beginning, there was only a request for application capable of risks evidence. Then we particularized the assignment iteratively and came to requirements following later on.

Company needs a new system for editing and evaluating project risks. The application will be used by multiple users working for the company. Each of the users can work on one or more projects. Therefore, he needs access to the records of these projects. New projects in system are created by project leaders. Each project has its own list of risks suggested by users working on the project and defined by project leaders. Risks in system are represented by their name, phase of origin, probability

of occurrence and severity of consequences. For each risk system must be able to calculate score evaluation based on these values and classify the risk to suitable class depending on its score. After classifying the risks, system must be capable of presenting and storing the resulting values for future usage.

Risks in projects are re-evaluated, so the list of risks of a project must be editable in time. System must especially support adding new risks to the project as they arise. Also the probability value and consequences of each risk must be editable in time. Current project phases can be entered as well. As the time proceeds, users can update the attributes of each risk in project, so that they reflect the real situation. System must be able to re-compute the score and display the results, risk classification and also evolution of these values in time. Previous values of attributes are to be saved for the purpose of illustrating development of risks in time. Most important feature of application is editable risk list and ability to recalculate the risk situation of the project based on updated risk attributes. Displaying development of project risk situation in time makes the project team capable of managing the project more safely and contributes to successful project finishing.

6.2 Requirements analysis

Based on specification interviews with Mr. Fiedler, informal specification of the application was transferred into formal, more detailed requests. They are divided into logical parts and each request is characterised by several key points:

- request label
- request body
- design and implementation priority (1-5, 5 is highest)
- implementation difficulty (1-5, 5 is highest)

6.2.1 Functionality

6.2.1.1 Projects

Individual risks must be assigned to particular projects that they influence. System must provide aggregate view on all risks belonging to a project.
Priority: 4
Difficulty: 2

6.2.1.2 Risk description

Application must provide means for describing all risks threatening the project. The output should be a list of possible risks.
Priority: 5
Difficulty: 3

6.2.1.3 Main risk attributes

Each risk will be described closer by its main attributes – probability of occurrence and consequences. These attributes must be assigned credible values with satisfactory accuracy level, so they may be used as a base for further calculations.

Priority: 5
Difficulty: 5

6.2.1.4 Risk evaluation

System must be capable of calculating a score value for each risk and assign the risks to different classes according to their score values.

Priority: 5
Difficulty: 2

6.2.1.5 Re-evaluation

Application must support re-evaluation of risks in time. After re-evaluation of risks' probabilities and consequences, their attributes must be updated and saved in system. When key parameters of risk are changed, system must allow recalculating their scores and reclassification of risks. System must support this recalculation as one of the key features of the application.

Priority: 5
Difficulty: 4

6.2.1.6 Old values

When re-evaluating the risks, the old values must be always saved and stored in database to be used later on for examining older results and making comparisons.

Priority: 4
Difficulty: 2

6.2.1.7 New risks

System must allow users to fill in additional risks identified during re-evaluation in different stages of project lifecycle. These risks must be taken into consideration from that moment on.

Priority: 5
Difficulty: 2

6.2.1.8 Evolution in time

System must provide means for demonstrating evolution of project risks in time as one of the key features of the system. System must provide data about risk increase and decrease in through the project lifecycle. System must provide this information at least in text, possibly also in graphical form.

Priority: 5
Difficulty: 4

6.2.1.9 Project stages

System must allow setting current project stages, so that the occurrence of risks through these stages is recorded and can be examined.
Priority: 4
Difficulty: 4

6.2.1.10 Printing output

System must be able to print output of the project risk situation analysis. Output will be printed in graphical or text form.
Priority: 3
Difficulty: 2

6.2.2 Users and human factor

6.2.2.1 Multi-user application

System should be composed as multi-user system and way of its service on target platform should reflect bigger amount of users working with the application through project teams and leaders.
Priority: 4
Difficulty: 3

6.2.2.2 User differentiation

Users interacting with the application work on different positions and places, system should therefore manage access to business information, knowledge and intelligence. This should be accomplished through limited access to projects' data and defining ideally 3 levels of user access to the application (technical support, common and superior access).
Priority: 4
Difficulty: 2

6.2.2.3 User access

Application will implement logging of the users (with login and password and) 3 levels of user rights.
Priority: 4
Difficulty: 2

6.2.2.4 Easy-to-use

Application should have intuitive and simple user interface, implement easy-to-use and user-friendly environment so that it is not difficult to switch to it from formerly used software.
Priority: 2
Difficulty: 1

6.2.2.5 Editing user profile

Application must provide possibility for users or at least for administrators, to update key information contained in user profiles, change key features like name, position or password.
Priority: 4
Difficulty: 3

6.2.2.6 User management

System must allow user management – means to add new users, display or edit information about existing users or deleting old users that are no longer working with the system.
Priority: 3
Difficulty: 3

6.2.2.7 Authorities

System must implement authority differences when managing the users or projects. Based on authority level, the access of a user must be restricted either to personal and technical data or to business data. Moreover, mainline users can have only read access to projects, only project leaders are capable of CRUD operations over the projects.
Priority: 3
Difficulty: 3

6.2.3 System interface and resources

6.2.3.1 Platform

Application must be capable of using on platform that will make it available and accessible from wide range of locations through the company. This follows from the concept of using the application by project teams containing larger groups of members. After its deployment, application must be reachable at least within the company's intranet.
Priority: 5
Difficulty: 4

6.2.3.2 Company resources

Application should be running on intra-company resources so it is easy to manage, always reachable from the company's environment and confidential information are not to be sent through outer network.
Priority: 5
Difficulty: 4

6.2.3.3 Simple to launch

Company requires application to be usable without installing anything on users' PCs. Furthermore, application needs to be easily running, not consuming too much IT resources.
Priority: 2
Difficulty: 3

6.2.3.4 Simple and reliable inputs

Input information readings for the application must be simple but reliable quantities, easy to measure, verifiable and robust enough to base project management and estimations on them.

Priority: 2
Difficulty: 2

6.2.4 Documentation

6.2.4.1 User guide

It is necessary to compose a User Guide document describing the way of working with the application for new users or users converting from using different software.

6.2.4.2 Programming documentation

Provide basic programming documentation of the application for the purpose of possible future application extending.

Priority: 2
Difficulty: 3

6.3   Use-case model

Use case model for the application (Fig.1), based on specification and requirements analysis was created to cover all necessary application functionality.
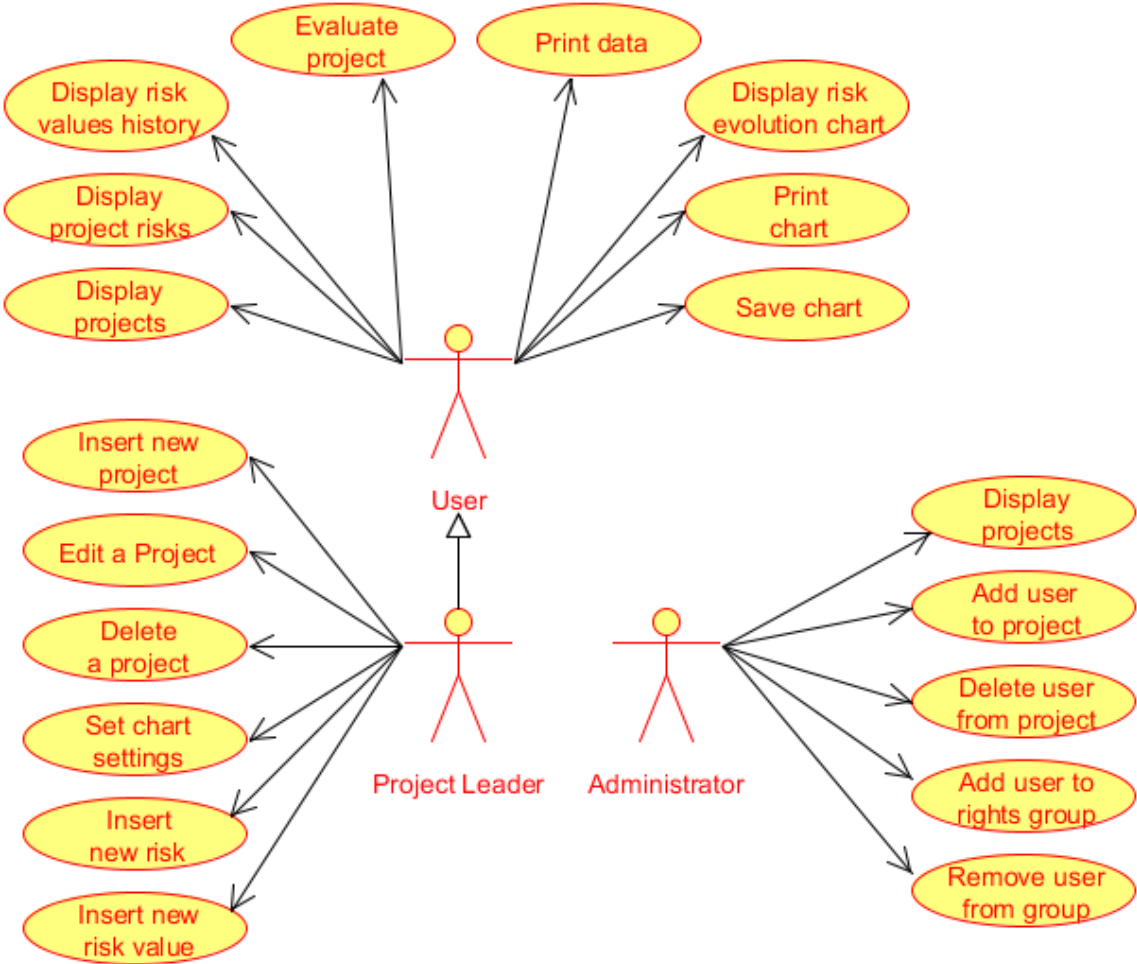
Figure 1. Use-case model of the application.

6.3.1   Actors

Following actors take part in relationships in designed application:

- User – is a standard representative of an application user. Has the common level of user read rights and is capable of providing common system functions. Main part of his activity in system is accessing, reviewing, processing and presenting attributes of projects and risks managed by the application. Has access only to projects he is assigned to.
- Project leader – has a superior level of rights in system and wider range of activities. Besides inherited activities from employee level he has different level of rights allowing him to add new records, edit and delete attributes of projects whose risks are managed through the application. Project leader is responsible mainly for adding fresh data into application

database and for project-related kind of work, but he can also work on the same activities like regular employees.

▪ Administrator – has different scope of work. Administrator is responsible for managing system users. He does not have access to customer's data related to project risks.

6.3.2 Use-case descriptions

Description of individual cases of use provides more detailed view of the application and its behaviour. Characteristic cases of application use are described in detail in Attachment A.

6.4 ER diagram

Entity relationship diagram (Fig. 2) describes the data dependencies in the application database. Table 'projects' stores the main application records of individual projects. It has a primary key defined on

Figure 2. Entity relationship diagram.

the 'project_ID' column. Table 'rights' defines the access of users to projects. For each user having access to a particular project there must be a record in 'rights'. The column 'project_ref' from the 'rights' table is a foreign key to the 'projects' table. Table 'rights' has primary key column 'id'
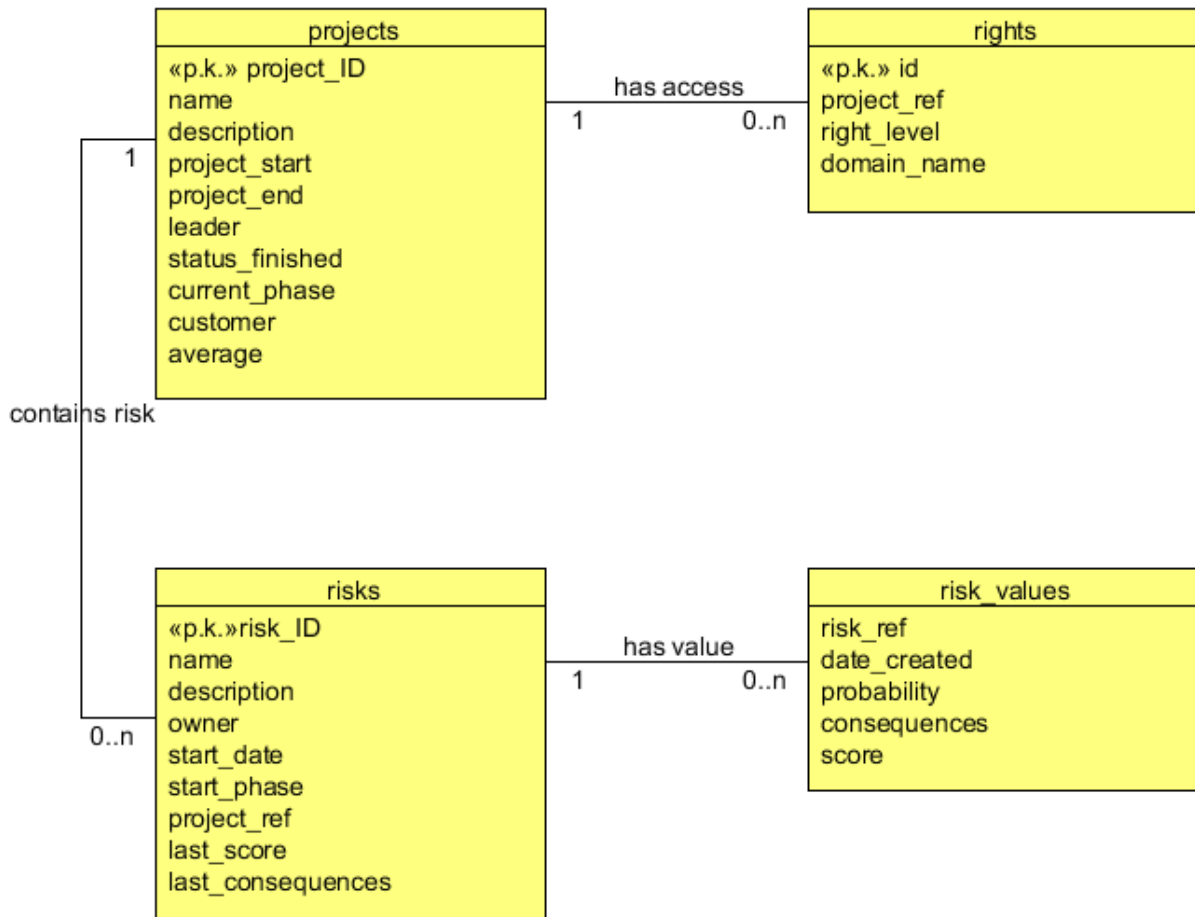
defined, to unambiguously identify the necessary table row with requested user access to a particular project.

Each project record can have virtually unlimited number of risks defined in the 'risks' table. This table stores the description of recorded risks for projects. Each table record contains a 'risk_ID', which is an auto-generated table primary key, so that each risk record has a unique identification. Records of individual risks contain their descriptive values, while all the history of risks probability, consequences and score readings is stored in 'risk_values' table. Records in this table contain history of measured values of numerical risk attributes for each risk. The values history for a particular risk can be selected by retrieving risk values by the requested 'risk_ID' and then sorting them according to the 'date_created' column.

7 Application implementation

7.1 Selected implementation environment

The implementation environment for the application had to be chosen in the view of the fact that the Allium company has already its own standards, platforms they use for developing and running their business applications and with respect to the licences for these platforms as well. Allium is a Microsoft partner, so using of Microsoft platforms and applications is natural and this trend can be expected in the future as well. Therefore, the application should be implemented on a platform that is Microsoft-based and the company already has it installed and running in daily use. Different platform would comprise additional expenses and effort to bring a new, diverse solution into common use.

Another requirement to consider was the fact that the risk management support application was supposed to be used by multiple users within the company. For these users, the cooperation on their projects is substantial, so the application was required to be implemented on a platform that allows sharing the resources and project data without a need to migrate and update project data to users' computers or to install anything on user's PC when moving to a new project.

In addition, the application must store the projects data in a transparent and common way, so that it offers an option of being interconnected with other company systems in the future. If the application, while testing, proves its contribution to the company's profit, it can be used for assessment of risks in their future projects.

Last but not least requirement for the application was to be user-friendly. It was supposed to be easily learnt by new users, with intuitive and easy-to-use interface, preferably implemented to be used in a well-known environment. This should make the testing phase and possible future usage simple and convenient for end-users.

With respect to all these requirements and their fulfilment, the following implementation resources were chosen:

- The database for the application was created in Microsoft SQL Server 2008. This is one of the current high-end data platforms offering extensive possibilities and great performance. It is easily manageable through SQL Server Management Studio allowing management of bases of

data via graphical user interface. This solution was developed by Microsoft and therefore is greatly compatible with other Microsoft-based solutions and environments.

- Main application platform itself is required to be used by multiple users in the same time, allowing sharing projects data in the intra-company environment. The most appropriate platform for this purpose is Microsoft SharePoint Server. This platform offers endless possibilities in sharing data, users' cooperation and business support. Hosting the application on server allows centralised database, better administration, accessibility from the inside and outside of the company and faster access to up-to-date information for users. Moreover, Allium is already using this platform for their business work, so there is no need for installing additional environment, the application will be just deployed to company's SharePoint server.

- User side – as users are connecting to MS SharePoint Server, there is no need to install any client application. The user application interface and shared content is available through MS Internet Explorer, which is MS-based again, so the compatibility is ensured. This way, full functionality of the application is available on user's computer, wherever he is working.

7.2 Implementation tools & requirements

Before application development itself, the proper system environment had to be prepared. The main platform, MS SharePoint Server 2010, requires 64-bit Windows operating system, so Microsoft Windows Server 2008 R2 Standard was installed and configured. Then Microsoft SQL Server 2008 Standard was installed and subsequently MS SharePoint Server 2010. As main tool used for designing of the SharePoint application was chosen and installed MS SharePoint Designer 2010. It was the most suitable tool for creating and configuration of vast majority of application elements. Finally, Microsoft Visual Studio 2010 was installed, since it supports MS SharePoint projects as well.

7.2.1 Microsoft SharePoint Server 2010 [13, 14]

As the base platform for implementing the application was chosen Microsoft SharePoint Server 2010. The first ancestors of MS SharePoint were two Microsoft platforms: SharePoint Team Services and SharePoint Portal Server. Later on, SharePoint Team Services and SharePoint Portal Server were combined in SharePoint 2003 platform. Next generation was represented by Microsoft Office SharePoint Server 2007 platform. The last generation of SharePoint platform, which has been improved a lot through its evolution, brings MS SharePoint Server 2010. It has a number of new features, better environment governance and reflects users' feedback from the previous generation.

MS SharePoint Server 2010 is a Microsoft platform for enterprise cooperation which has been attracting a lot of attention in last years. It was developed to be customized to theoretically each and every organisation with features supporting numerous business needs. SharePoint allows connecting people, sharing content, information and resources, supporting intra-company information sharing, managing relationships or projects with business partners and publishing information for customers. It is also a content, web and resource management system for companies, enabling wide range of information, resources and knowledge to broad user audiences through web services. Also, it provides extensive features for creating custom solutions for specific company's needs and decision supporting.

It unifies collaboration, sharing, managing enterprise content, business workflows, business intelligence, enterprise search and portals effectively in professional manner. If required, it can provide all of the above at the same time.

Altogether, SharePoint is a multi-architecture platform that is highly customizable and can support any business area and demand. According to the Microsoft [15], the most important features of the MS SharePoint Server 2010 are the following:

- **Portal server with distributed administration** – end users with no experience or skill in designing or administering web sites can easily create and manage their own sites with custom functionality and engaging appearance. This especially enables creating team collaboration pages, user blogs, discussion boards, forums and other information sharing sites.
- **Groupware applications** – for wider collaboration, SharePoint directly supports sharing team workspace, documents, workflows, themed lists or data sources.
- **Workflow hosting** – workflow support is a strong feature for formalizing, representing and managing enterprise processes.
- **Different content type management** – documents, records and web contents can be stored and managed in a single solution.
- **Business Intelligence toolset** – provides components for integrating external data sources, integration tools and user interfaces for end-users. External resources can be accessed as if they were SharePoint structures. Fundamental Business Intelligence tools supported are: Integration and Analysis services, Excel services for access to spreadsheet data and reporting, Visio, Performance Point, Reporting services and Excel Power Pivot for creating of custom user Business Intelligence solutions.
- **Complete intranet solution –** the entire enterprise intranet solution can be implemented in SharePoint, which can serve also for file storage and copying, host another applications and services, data storage and security and offer other services that an intranet solution may require.
- **Service hosting** – MS SP 2010 provides features for hosting a vast array of services. The client object model is used to access the data, along with the Windows Communication Foundation Data Services and various so called "out of the box" web services.
- **Storing data –** SharePoint uses its own data structures for storing data, SharePoint Lists. These are physically stored in a MS SQL Server database. They can be manipulated through LINQ or CAML language. Managing data also includes distributed data handling operations offered by underlying SQL Server edition.
- **Different levels for different interfaces** – except for the default web page user interface, there are additional levels supported by SharePoint. Mobile devices are provided a separate level of web page interface. Different interfaces are implemented for MS Silverlight and MS SharePoint Workspace applications. SharePoint is also accessible via Windows Presentation Foundation or Windows Forms thanks to the SharePoint client object model.

These are only some of the main features, as SharePoint in general is a very robust and multilateral platform.

Edition used for implementation: Microsoft SharePoint Server 2010 Enterprise

7.2.2 Implementation tools

Microsoft SQL Server 2008 [16, 17]
MS SQL Server 2008 is a recent Microsoft productive and intelligent data platform designated for small to mid-range data storages. The platform represents scalable, reliable and highly secure base for applications and offers enterprise data management with wide range of integrated services. It also supports connection to custom applications using or developed in MS Visual Studio, Service Oriented Architecture, MS BizTalk Server or MS Office products.

Edition used for implementation: Microsoft SQL Server 2008 Standard

Microsoft SharePoint Designer 2010 [18, 19, 20]
MS SharePoint Designer is a free, advanced developing tool for designing complex SharePoint solutions, rich and robust applications, web sites and pages, workflows and other features that SharePoint provides. This design tool offers various kinds of features to build and deploy progressive custom solutions for MS SharePoint non-programmatically. But if desired, a developer can still see and edit the generated XSL (eXtensible Stylesheet Language) code, observe the changes he has done in real time and thus customize the solution to reach a desired form and functionality.

The fundamental relationship between SharePoint Server 2010 and SharePoint Designer 2010 is following: SharePoint Server is a large, mighty, complicated web application that can contain, store and process theoretically any kind of content and is scalable to potentially any business case. SharePoint Designer, on the other hand, is an extensive, sophisticated design tool, implemented as a desktop application, with powerful capabilities for editing and customizing the way SharePoint content is presented, gathered, processed or exported. That means, SharePoint Designer is not used to modify the SharePoint content itself.

SharePoint Designer allows visual development using assembling the solution from custom or out-of-the-box components and visual web parts and custom or pre-defined controls. Main functionality includes creating, adding and customizing master pages, site pages, page layouts and cascading style sheets, creating and customizing workflows modelling business processes, adding and customizing SharePoint structures like lists, libraries and Data views and tools to completely control or customize the user feeling while working with the built application. Together with connecting created solution to different internal or external data sources, facilitating communication between both internal and external objects and components, implementing custom external content types and data operating functions, developers are provided a cutting-edge ability to create highly-profiled SharePoint applications on the top of SharePoint platform, precisely according the business needs. And finally, one of the biggest benefits SharePoint Designer brings at all, is the so called "WYSIWYG" (What You See Is What You Get) development environment, enabling developer to visually adjust the built solution. This way, the designer has a perfect overview of developed solution. SharePoint is even capable of retrieving the representative sample of data from database to populate the tables, lists, etc. The WYSIWIG feature is also invaluable when creating or editing custom user interfaces and overall solution lookout.

Unlike the other sites, SharePoint sites do not store site content in particular MS SQL Server databases. Therefore SharePoint Designer is not accessing the site content directly when editing it. Instead, the content is loaded and modifications are saved to the database solely through SharePoint Server (or SharePoint Foundation) and SharePoint Designer is only accessing the site through a URL link the same way as a classical web browser would do. Hence working with SharePoint Designer implies a necessity of being connected to an environment with running SharePoint Server or SharePoint Foundation.

Edition used for implementation: Microsoft SharePoint Designer 2010 (x64)

Microsoft Internet Explorer 8

Microsoft default web browser Internet Explorer 8 or higher is fully compatible with SharePoint 2010 sites. Some components of the SharePoint solutions may require the recent Java and Flash updates installed on the client's side.

The importance of Internet Explorer is increased due to the fact that it allows editing of pages - some of the web parts, from which a SharePoint page consists, can be created, deleted and updated to some extent. This means that a SharePoint page can be basically edited through a simple browser view. Also the security attributes of the page and part of its administration settings can be updated through web browser.

Compatibility of some of the lower versions of Internet Explorer may be sufficient, but cannot be guaranteed. Web browsers like Mozilla Firefox or Opera turned up not to be entirely compatible with created SharePoint solution.

Edition used for implementation: Microsoft Internet Explorer 8 (x64)

Microsoft Visual Studio 2010 [21, 18]

MS Visual Studio was utilized when creating solutions for some of the custom web parts and deploying them. Visual Studio 2010 already includes several templates for solutions for SharePoint 2007 and SharePoint 2010 as well. There custom solutions can be built, deployed into the local SharePoint Farm and custom web parts can be inserted into SharePoint sites afterwards. This can be particularly the case of web parts like chart web parts, custom data views or external lists and external content types definitions. MS Visual Studio 2010 may be also used to deploy Silverlight web parts into the SharePoint Farm.

Edition used for implementation: Microsoft Visual Studio 2010 Professional (x64)

7.3   Creating application database

7.3.1   Database creation and initial settings

The first step after specifying requirements for implementation and choosing implementation tools was installing the selected data platform – Microsoft SQL Server 2008 Standard edition. After successful installation and quick initial settings, Microsoft SQL Management Studio was used to prepare the data platform environment and create a base for the application data.

The database was created on a local instance of SQL server named "WIN-SH9N3UGAOIE". The name of the database was set to "Rizika". The owner of the database became the default user on the Windows Server 2008 system, "WIN-SH9N3UGAOIE\Administrator". This way, database was owned by a user with the administrator rights and set up to run under the local server "WIN-SH9N3UGAOIE\Administrator" account. For the case of future deployment of the database into running production environment, it would be appropriate and more secure to have the database running under a special system account with its rights set accordingly, in compliance with the IT standards, owner requirements and security standards of the owner.

For connecting of the application to the database using SQL authentication, an account with the name "risk_application" was created. The permissions for this account were set accordingly to the application needs, with the effective permissions to "CONNECT", "SELECT", "INSERT", "UPDATE", "DELETE" and "VIEW DATABASE STATE". No permissions for changing design, hierarchy, status or settings of the database were granted. These permissions are available only to database owner.

7.3.2   Design of data tables and views

Four data tables along with one data view were created for storing necessary types of application data records. Description of attributes of each table follows.

Table [projects]
Basal for the application is a table with records of monitored projects. The table contains columns of following names and types:

Column name	SQL data type
[project_ID]	bigint
[name]	nvchar(50)
[description]	nvchar(MAX)[2]
[project_start]	date
[project_end]	date
[leader]	nvchar(50)
[status_finished]	int
[current_phase]	int
[customer]	nvchar(50)
[average]	numeric(18,0)

Table 6. Column names and types of SQL table [projects].

Column [project_ID] contains the ID of a project record. It is set to automatically increment when adding new project record into database. This is also a Primary Key column of table [projects]. Columns [name] and [description] contain information about name and description of a project. Their maximal length is noted in Table 6. Columns [project_start] and [project_end] represent starting and

---

[2] The maximum storage size of data type NVARCHAR(MAX) is 2^{31}-1 bytes, being equal to 1 073 741 822 Unicode characters [22].

expected ending date of a project. Column [leader] contains a name of a project leader. Columns [status_finished] and [current_phase] contain integer values referencing project state and current project phase from the set pre-defined project states and pre-defined project phases. Column [customer] contains name of the project customer and column [average] states the numerical value of weighted average of current project riskiness.

Table [rights]

Table [rights] contains information about user access rights to particular projects. Table contains four columns: column [id], which is an auto-incremented column containing ID of each record in the table. Column [project_ref] is a Foreign key referencing ID of a project, to which the respective record belongs. It is a foreign key to the table [projects] referencing [project_ID] column. Column [right_level] contains integer value from pre-defined set of values of user right levels for a particular project. Column [domain_name] contains a user domain account that will be used by the user to log into the application.

Column name	SQL data type
[id]	bigint
[project_ref]	bigint
[right_level]	int
[domain_name]	nvchar(50)

Table 7. Column names and types of SQL table [rights].

Table [risks]

Information about risks occurring in particular projects is stored in [risks] table. The auto-incremented identifier of each risk is stored in [risk_ID] column. Name and description of the risk is stored in columns [name] and [description]. Each risk has to have an owner, his name is contained in [owner] column. Columns [start_date] and [start_phase] contain the date of risk origin and a pre-defined value of the parent project phase in which the risk originated. The ID of the parent project is found in [project_ref] column, a Foreign key to the [projects] table. Column [last_score] contains the actual value of project score and [last_consequences] is an auxiliary column bearing the value of actual consequences of the risk.
Table [risks] has a Primary key hosted by [risk_ID] column.

Column name	SQL data type
[risk_ID]	bigint
[name]	nvchar(50)
[description]	nvchar(MAX)
[owner]	nvchar(50)
[start_date]	date
[start_phase]	int
[project_ref]	bigint
[last_score]	int
[last_consequences]	numeric(18,0)

Table 8. Column names and types of SQL table [risks].

Table [risk_values]

The [risk_values] table contains recorded values of individual risks which are used to compute the current risk score. Table has five columns: [risk_ref] is a Foreign key to the [risks] table referencing parent risk of the particular value record, [date_created] is a date when the values of the risk were determined. Probability of risk on this date is stored in [probability] column, estimated consequences value of the risk is in [consequences]. When user submits these values, then the [score] value is computed as a product of the [probability] and [consequences] values. The values of [score] and [consequences] are then updated also into [last_score] and [last_consequences] columns of the particular risk record in [risks] table.

Column name	SQL data type
[risk_ref]	bigint
[date_created]	date
[probability]	numeric(18,0)
[consequences]	numeric(18,0)
[score]	int

Table 9. Column names and types of SQL table [risk_values].

View [project_rights_view]

This view selects data from three database tables – [projects], [rights] and [risks]. Columns selected from [projects] table are: [project_ID], [name], [description], [project_start], [project_end], [leader], [status_finished], [current_phase] and [customer]. From table [rights] were selected: [id], [right_level], [project_ref] and [domain_name]. The [risks] table has no selected columns, but is used in an aggregate calculation of weighted average of riskiness for each project that has the values of its risks defined. The [consequences] value of each risk is used as its weight criterion, the averaged variable is risk [score]. This calculated value of project riskiness is stored in a view column [w_avg].

For the purpose of retrieving all required data in correct relationships, tables [projects] and [rights] are connected using INNER JOIN command on condition [projects].[project_ID] = [rights].[project_ref]. The whole result set is then connected with [risks] table using LEFT OUTER JOIN on condition [projects].[project_ID] = [risks].[project_ref].

Using JOINs allows the application to determine correct access of users to projects and to compute the weighted average of riskiness for each project in database that has values of risk filled in. The query for creating the view is stated in Attachment C.

Table and view dependencies

Once all the tables and views were created, we can recognize certain hierarchy and data dependencies throughout the created database. The uppermost table in the hierarchy is the [projects] table. Tables depending on [projects] are [rights] and [risks], because they are both containing foreign keys to [projects]. That classifies tables [rights] and [risks] to the second level in table hierarchy. Table [risk_values] is dependent on [risks], as it contains a foreign key referencing [risks] table. That ranks the [risk_values] to the bottom of table hierarchy in created database.

Another kind of dependency exists between database tables and view [project_rights_view], which is selecting the data out of joined tables [projects], [rights] and [risks]. The view uses first two tables for selecting data and the third one to apply aggregate functions on selected table rows.

Sample data

After definition of database tables and a view, they were filled with several rows of sample data to verify correct functionality and proper joining of tables. The select, update and delete queries were used to verify the hierarchy and dependencies of database objects. The sample data also served for debugging and testing of functionality during the development of application itself.

SQL scripts

The SQL scripts for creating application database tables can be found on enclosed DVD in the folder "/*database*"

7.4 Creating SharePoint site for the application

After installation of Microsoft SharePoint Server 2010 it was possible to create an application site on the local SharePoint Farm. The most appropriate way to create a new site was to connect to the Home page of the SharePoint server using a web browser and create a new site on the SharePoint Server farm. The address of a SharePoint Server's Home is either "http://<"SharePoint-server-name">/" or "http://localhost/". Both of these addresses cause the Home page of a correctly installed and configured SharePoint server to load. The picture (Pic. 1) shows the loaded Home page, already after a slight modification

.
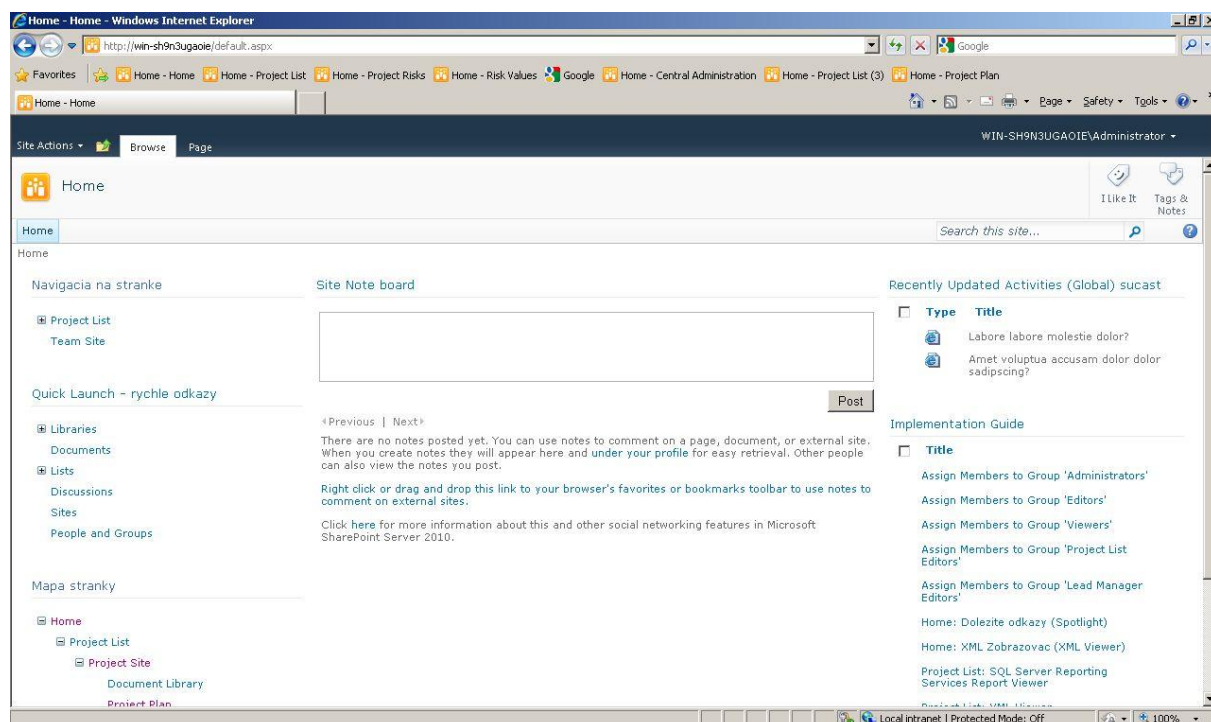
Picture 1. The server Home page is loaded.

In the upper part of the Home page there is a wide menu panel with available functions called a ribbon. At the left side of the ribbon there is a button "Site actions" that contains the main actions available for the central SharePoint Server home site. From the menu a "New Site" item was chosen and the window for creating new sites appeared. The site was named "Project List" with the URL name "ProjectList" on the local SharePoint server. When creating a new site, SharePoint offers a wide range of site templates (Pic. 2), from which the new site can be based. These templates are reflecting the most common business needs. For the purpose of my application, however, it was chosen to create the solution from a Blank Site template. After creating a site for the application, it was necessary to create a new page for the application implementation. The web browser was used to navigate to the Project List site that was just created. From the Site Actions menu, "New Page" was chosen and a name entered for the new application web page. A new page was displayed in the edit mode. In spite of the fact that SharePoint allows creation, editing and management of pages through plain web browser, it was more useful just to click the "Save and Close" button, leaving the page blank. The new web page was added to the site pages library. The reason for this was that the browser is suitable mostly for simple actions related to sites and pages creation, editing, management or removing. SharePoint, of course, allows much more sophisticated and complex actions and customizations of the solution to be done through the browser, but it is far more convenient to use more advanced development tools for designing and configuring the solution.
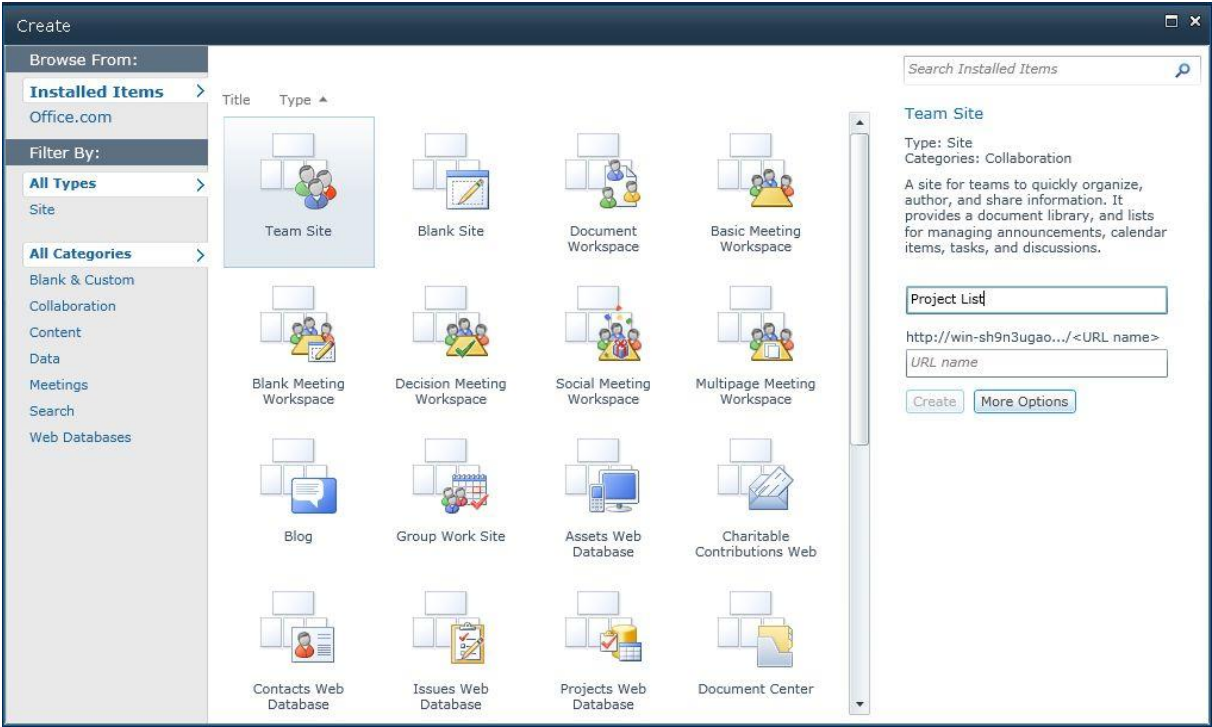
Picture 2. New site template can be picked up.

The main such tool for creating complex and sophisticated SharePoint solutions is naturally Microsoft SharePoint Designer, which is exclusively designated for no-code development on the SharePoint platform. Since the solution in SharePoint Designer is built from the pre-defined components, it is a simple and convenient method of creating solution for users without extensive skills in SharePoint development or web design. Second advantage of developing in SharePoint Designer is the time

saving aspect, as without the need to code the entire solution, development becomes several times faster than before.

Another sophisticated design tool supporting SharePoint development is Microsoft Visual Studio. Although there were some extensions for SharePoint available also in Visual Studio 2008, the latest version, Visual Studio 2010 supports SharePoint directly having pre-defined templates for SharePoint 2010 and SharePoint 2007 projects in different programming languages. Visual Studio is a powerful design tool where required solution functionality and shape can be programmed to the last detail. It is therefore implicitly predestined for creating custom components and functions of the solution and used mainly by experienced and skilled SharePoint developers.

Central Administration and Site settings

Before starting with further development in a more sophisticated tool, there is one more important feature of the web browser to facilitate. The SharePoint Central Administration page is accessible solely through the web browser. The same applies to the Site Settings page of each site. Central Administration page is accessible via a link from the folder Microsoft SharePoint 2010 Products in Windows Start menu. Central Administration manages features for creating web applications and site collections, managing service applications and deploying solutions, changing system settings, set up monitoring, defining security settings, performing backups or restores, upgrading, migration and adjusting general application settings. It can also contain configuration wizards.

As it was necessary to assure the correct application settings for the developed solution, the section Manage Farm Features in System Settings group was reviewed. The list of adjustable SharePoint farm features was listed and checked to be sure that features "Data Connection Library", "Excel Services Application Web Part Farm Feature" and "Global Web Parts" were set to Active state. After setting the necessary Farm features, the Central Administration site was navigated again and section Manage service applications in the Application Management group was entered. It was necessary to ensure that the Business Data Connectivity Service, Excel Services Application and PerformancePoint Service Application were started, as they support features used in further development of the SharePoint application. After preparing the SharePoint environment and checking the necessary settings, the development in SharePoint Designer 2010 could start.

7.5   Developing the application in SharePoint Designer 2010

A new site for the application had already been created, so it was necessary to create the underlying structure for modelling the application hierarchy. After launching SharePoint Designer the Project List site was opened using the background menu. Site properties were displayed in the main part of SharePoint Designer window. This window enables to change main settings of the site: site's name, description, web server folder, allows customization of site's home page, changing site themes, few general settings, setting site permissions and managing subsites. On the left side resides a Site Objects panel. It contains libraries with different kinds of site objects. The last item of the panel is All Files. After clicking on it, a list of site folders and files was displayed. Amongst them the page created using

the web browser was selected, its file right-clicked and set as homepage to become the default page for the site.

Application structure

The structure of application was designed to reflect application hierarchy, as follows: The top level of application hierarchy will consist of the main application site called "Project List". It will be created in the local SharePoint Farm. The site will contain pages related to managing of project records. The default application page named 'default.aspx' will contain the list of projects available to the user viewing the page, with respect to his access to projects and level of user rights. There will be also an administration page 'Admin.aspx' for managing user access to projects. It will be available only for application administrators.

The middle and bottom application levels will be hosted on a created subsite named "Project Plan". The subsite will contain the page 'default.aspx' which will be serving for project risks displaying and addition. It will represent the middle level of application hierarchy. The bottom level will be formed by the page 'risk_values.aspx', which will contain information about risk values history.

All of the application pages will consist of deployed web parts and a ribbon that is inherited from the master page template of the site. The look of individual web part pages will be described in the subsequent chapters. Dividing the application levels into subsites, instead of using a single central site or even root SharePoint Farm level only, makes the application structure more transparent and organized. It also creates favourable conditions for both future application extension and deployment of application into real business production environment.

Communication of necessary data variables between individual application pages is implemented through passing these variables in URL parameters by the sending page. The web parts of the receiving page that are expecting data retrieve them by parsing the QueryString. Expected data are then bound as web part parameters and can be used in the application. The main data sets have to be retrieved from Microsoft SQL Server database, so for each web part working with external data there is a data source defined based on a database connection for the particular database object. These data are then presented through Data view web parts. For feeding data to Chart web part, it may be necessary to define External Content Type to retrieve the content from a database. From an External Content Type the data can be populated into an External List. Finally, the External List can be used as a data source for the default chart types.

The Ribbon

The upper Ribbon user interface on site pages is inherited from the Master Page site template, from which all other site pages in SharePoint infrastructure are generated. It improves working with page design, web part properties and shows contextual tabs with controls for the on-clicked page elements.

Web parts

SharePoint web parts are out-of-the-box independent components, from which SharePoint pages are built. SharePoint offers numerous web part types for different purposes and applications. They can be differently aligned on the page, customized to have the desired functionality, to view data from external or SharePoint data sources or to have another function. The Data view web part will be the most suitable type to choose, because the application data will be retrieved from a database.

SharePoint provides a variety of themed web parts as well, but for our purpose the Data view will be inserted as an Empty Data view and customized afterwards.

Differencing Data view functionality

The determining criterion for differencing the availability of site features to the users is the user group that the features are intended for. Therefore three levels of application component functionalities were created, corresponding to three levels of user access to the application features and project data. Each web part in SharePoint Designer has an attribute called 'Target audiences' that represents the user audiences allowed to access the web part on the web part page. If various access levels are applicable, the web parts are created in three different versions, each providing diverse functionality to one particular class of users. On the other hand, if 'Target audiences' attribute of a component is undefined, then no access restriction apply to it. This way it is possible to differentiate application content and functionality to be accessible only to intended users or user group members.

First access class is a basic level of user access designated for common users of the application. At this level, users can access the web page components providing basic functionality described in the following chapter. This level of access is, in fact, a standard Read access, meaning users are allowed to see the data of projects, risks and risk values, but cannot modify or delete them. Nor are they allowed to add new data into the application. Users with this level of access can display selected web parts with target audiences attribute set to "Site_users" group or undefined target audiences.

Second class of user access is designated for project leaders. They are members of user group "Site_projectLeaders". They have access to page components with extended functionality. Apart from having the basic functionality at their disposal, project leaders are also capable of CRUD[3] operations over the project records. They can also add new risks for particular projects and enter new values of these risks, because project leaders, which are in charge of projects, are expected to re-evaluate the project risks on regular bases and record the values in database. Editing or deleting risk values has no sense in the context of application requirements.

Third class of access is designated for application administrators, members of the group "Site_admins". These users do not have complete access to business information, they just have read access to project records. Their purpose is adding other users into the application database and assigning them into appropriate user groups to provide them with necessary level of access for selected project. "Site_admins" is the only group with the ability to grant or cancel membership of users in user groups. They can do so using the menu "Site Actions" – "Site permissions", where they can edit the membership of users in particular groups.

There is also a fourth level of user access, the respective group is named "Site_owners". Members of this group have full control rights to the whole site, its content and structure. This is typically one-user group for server administrator or other superusers.

Communication vs. security

With respect to the intended application use and the environment in which the application is going to be deployed, the implementation of user access to project records serves more to prevent users from viewing or editing unauthorized data unintentionally than to robust security enforcement. It is not security in its real meaning, because each of the users that are members at least of the "Site_users" group can theoretically view the web part pages with the data populated. This way it is possible, e.g.

---

[3] CRUD is an abbreviation for Create, Read, Update and Delete

through a modified URL, to view and access content of the projects for which user does not have authorization granted by site administrator. If the user had the project leader level of access, he could also possibly use customized URL link with spied parameter values to modify the data of another team's project. (If he was not member of neither of the "Site_users", "Site_admins", "Site_projectLeaders" or "Site_owners" group then he could access the pages but not the web parts containing data). But since the application is planned to be deployed in a small company (where most of the employees work together in the offices without any intension of misusing the application to cause any harm or damage) and the application is going to be accessible only through the company's intranet, this level of restricting unauthorized users from viewing the content is sufficient for the purpose of risk management demonstration. If the application was to be used on an extranet site, more advanced level of security would be necessary. However, that does not come under the topic of this thesis.

7.5.1   Creating the List of projects for a particular user

The list of projects available for a particular user was created on the default page of the application main site. This view is the starting point for the application user and corresponds to the highest level of application hierarchy. The source code of the page is contained in the file "/source/ProjectList/default.aspx" on enclosed DVD and a fragment sample is attached in Attachment B of this thesis. Within the SharePoint site collection, the page is located in "…/ProjectList/default.aspx".

Central Data view
Next step was designing the page itself. The design view of the page was open to create the design of the page using pre-defined and custom web parts and controls. As a site default page, it had to contain the list of projects available for user. The projects data were to be retrieved from SQL database, so a Data view web part was chosen as a base web part for this page. Firstly, a new data source had to be created in the Data Sources gallery of the site. After opening the gallery, a new database connection was added, named "projects on Rizika". After server name and login data for 'risk_application' account were filled, table [projects] from database [Rizka] was chosen as the source of data. Later, the data source had to be changed to the view [projects_rights_view] and it was named "project-rights", because the relationships between data records needed to be utilized for managing user access to projects for particular users. When the data source was created, it was possible to insert a central Data view web part to the page and connect it to data source. Firstly, empty data view web part was inserted and it was configured to connect to the "projects on Rizika" data source. Columns [project_ID], [name], [description], [project_start], [project_end], [leader], [status_finished], [current_phase], [customer] and [w_avg] were selected to be displayed in a multi-item view in the web part. After confirming the selected columns SharePoint Designer created a table populated with the data records loaded from the data source. All these steps could be done in the graphical design environment.

After its creation, the next step was customization of the inserted Data view web part. This was supposed to be achievable by displaying so called Common control tasks or other settings of respective elements and adjusting proper values of desired attributes of the object. Other way possible was editing the generated source code in the code view of SharePoint Designer. Many times editing

the source code promptly showed to be the best or even the only possible method to set the proper attributes and achieve desired functionality. Quite often the graphical menus for formatting and editing items did not show all possible options available or did not let the designer edit the desired property of web part element. This was the reason why it was necessary to work in the Split view (Pic. 3) of SharePoint Designer most of the time. Based on practical experience, the most convenient way of designing pages and web parts was adding the elements to the page in the graphical Design view and perform some of the main settings. Then it was necessary to switch to the Code window and make specific changes, settings and customizations. Despite SharePoint Designer being a WISIWIG no-code design tool, in vast majority of cases working in the Split view and editing the generated code was faster and easier than trying to set up everything in the Design view.
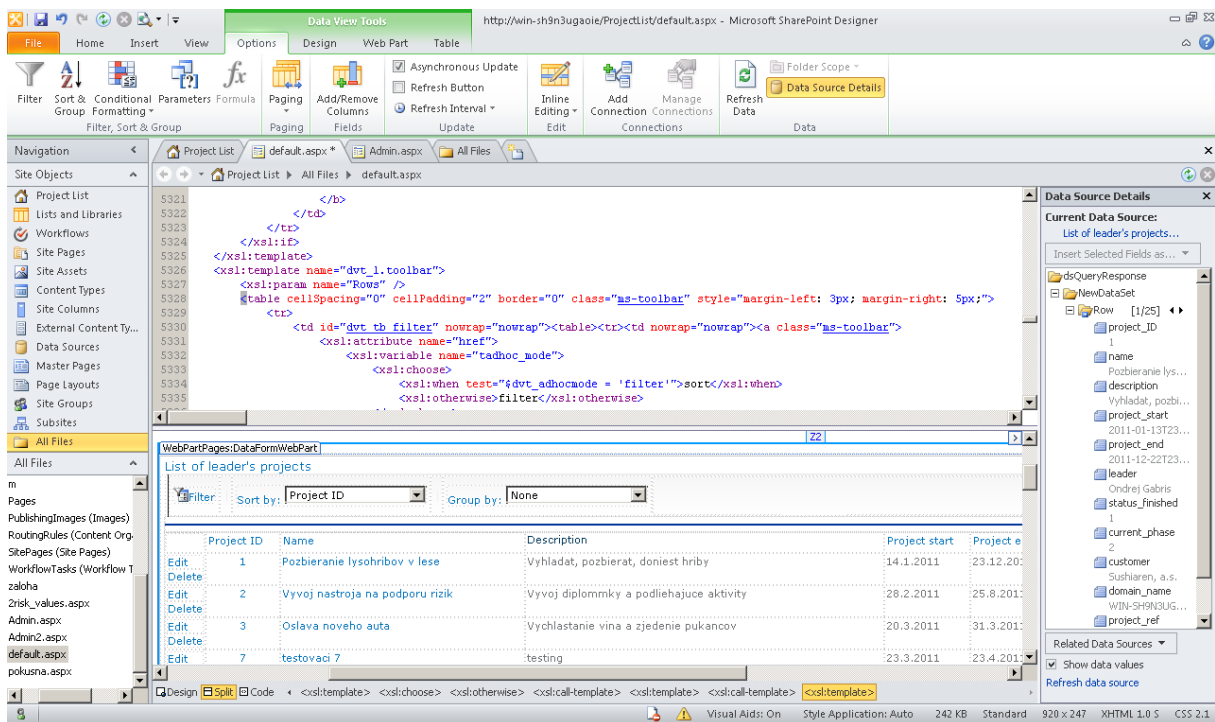
Picture 3. A list of user projects is being customized.

Most important customization after inserting the data table was setting correct data format for some of the columns. Especially columns containing dates had to be reformatted and to be better understandable, localisation of the date format had to be set. Secondly, column headers had to be edited to bear clear descriptions of the data contained. They were also set to be sortable upon a mouse click, what is a very useful and depicting ad-hoc feature greatly improving the user's feeling while working with application. Then data alignment was set within the cells and the table cells dimensions were adjusted to fit to the page. The project ID values and project names were added a hyperlink to the Project risks page (lower in application hierarchy), containing information about project risks. Information necessary for communication between the page containing list of projects and page displaying project risks was stipulated to be sent through the URL link as a parameter. Finally, a toolbar for filtering, grouping and sorting was added to the top of the Data view, so that the data could be filtered and grouped by selected criterion or sorted according to user's will. As stated above, all of these Data view features were added to the Data view in the design mode and then customized by

examining and editing the source code in order to respond to the application needs. After applying XSLT filter for displaying only the records of projects to which user has access and giving a Data view a suitable title, the basic concept of the Data view was finished.

Functionality extensions

Basic Data view concept completely fulfilled the functionality required for "Site_users" group members. For other levels of user access, however, apposite modifications or functionality extensions were needed. For site owner and project leader levels of access, the CRUD operations had to be implemented, particularly Insert, Edit and Delete operations. Therefore, an Insert and Edit mode for the original Data view concept was implemented, involving numerous source code modifications and additions. The Insert template of the Data view was created and input fields with proper labels were suitably fitted. An Edit template was implemented for in-line editing of project records directly in the Data view table rows. Data for project ID and average project riskiness stored in database columns [project_ID] and [w_avg] were not to be gathered through the insert template or modified in Edit mode, as they are evaluated automatically. The validators of input fields were implemented to validate user's input type or range (if necessary) before committing fresh project data into application database. An 'Insert' and a 'Cancel' links were created in the bottom of the template to commit new project data into application database. SQL logic was implemented to save the domain name of the project creator into the database, so that an author of project record would have access to it by default. As for an Edit mode, the 'Save' and 'Cancel' links were implemented to be shown at the edge of the edited table row. A 'Delete' link was placed aside each project record in the table as well. After clicking on 'Delete' link for a project, entire project record, including all its risks and risk history values, is deleted from the application database. The data view web part was allowed asynchronous updates, so it was not needful to refresh entire page every time the project record table was modified. This gives the solution an advantage which is highly appreciated in real production environment. Last thing remaining to be set up was the web part title – it was set to "List of leader's projects".

Another functionality extending was needed for users being members of "Site_admins" group. The basic Data view concept was usable for them, but hyperlinks comprised by IDs and names of projects were required to point to an administration web page allowing adding or removing user access for the selected project. The necessary information was to be passed to the administration page through URL parameters. The aforementioned administration page will be described later. Despite the title of the web part was set to "List of admin's projects", it is accessible also to the members of "Site_owners" group.

Navigation panel

"Site navigation panel" web part was created from a custom Data view web part by removing all the redundant components and source code. Instead, a simple table having a few rows was defined and labels for particular application levels inserted. As the web part is located in the upmost level of application hierarchy, only the highest label was designed as a hyperlink pointing to the same page.

7.5.2   Creating the user access administration page

The page code is stored on enclosed DVD in "/source/ProjectList/Admin.aspx". On the SharePoint site, the page can be found here: ".../ProjectList/Admin.aspx".

List of users with access to the project
Project access administration page was created at the same site like the starting project list page, because it concerns the projects, even though on a lower level of application architecture. A new source of data had to be created in site's Data Sources library to retrieve domain names of users with access to selected project from the [rights] table. It was named "rights on Rizika". Afterwards, a Data view web part was inserted and connected to the data source. The table of user domain names having access to the selected projects was created by the Data view web part. An ID and a name of the selected project was placed in the top section of the web part. These values are passed to the web part through QueryString variables from the main application page.

When it comes to the bare access administration (The question is: Allow or deny access?), everything that matters for the application logic is a unique domain name of the user accessing the project. Therefore this web part titled "User access for project" is listing domain names of users that have any level of access to the selected project. A specific level of access to the project is granted to the users according their membership in one of the "Site_users", "Site_projectLeaders", "Site_admins" or "Site_owners" groups.

To grant new user an access to the selected project, the links from "List of admin's projects" web part were pointed to this page, passing the parameters through the URL. After customizing SQL queries used for data manipulation, the table containing one column with user domain name was created. Data table is sortable by clicking its header. The links for removing particular users were created aside each table row. The link for adding new username was situated to the bottom of the table. When the link is clicked, an input field for entering new user's domain name appears together with links to add the user into the database or cancel the operation. User will not get desired level of access until he is added by administrator into one of the "Site_users", "Site_projectLeaders", "Site_admins" or "Site_owners" groups.

Navigation panel
"Site navigation panel" web part for this page was designed to carry link to the list of projects at the highest application level and a name of the currently edited project formatted as hyperlink pointing to the same page and containing ID and name of the selected project.

7.5.3   Creating project risks page

A new subsite called "Project Plan" was created at the main application site for the risks of a particular project at the address ".../ProjectList/ProjectSite/ProjectPlan/default.aspx". The page file is also available on enclosed DVD in "/source/ProjectPlan/default.aspx". Default page of the subsite contains the risks of a selected project and a page named "risk_values" shows history of values for the selected

risk.It is enclosed on DVD in "/source/ProjectPlan/risk_values.aspx", the page web address is "…/ProjectList/ProjectSite/ProjectPlan/risk_values.aspx".

Project risks

The default page of Project Plan subsite was designed to provide a detailed list of risks occurring in a project selected on the main application page. A Data view web part was used again, as the risks records for a certain project have to be retrieved from application database and displayed in a structuralized view. To achieve this, a new Data Source called "risks on Rizika" was created. It was set correct parameters for database connection and data retrieval from the [risks] table. Data view web part was connected to this data source and columns [risk_ID], [name], [description], [owner], [start_date], [start_phase] and [last_score] were selected to be displayed in the web part as a multi-item view. After selection, data records were filtered according to the ID of a containing project and table was populated. While these steps could be performed in graphical Design view, further customizations were better performable by editing the source code in the Code view of SharePoint Designer.

Customizing the Data view included mainly reformatting the data columns and setting proper localisation parameters, setting suitable column headers and making the columns sortable on a mouse click. Data within the cells were set suitable alignment. The ID value and the name of each risk were formatted as hyperlinks pointing to another page of the subsite, where risk values history for the selected risk could be examined. Information about selected risk and the containing project was to be sent to the page through parameters. In the end, a title for the web part was set.

Functionality extension

The basic concept of Data view web part showing risks of selected project is usable for the basic level of user access, as it just allows viewing of the data. This is sufficient for standard users from the "Site_users" group. For project leader access level, additional functionality was required, to allow them inserting new data into risk database. So the link for inserting data was added to the bottom of the Data view and an insert template for entering new data records was created (Pic. 4). Input fields with suitable labels were fitted for the necessary risk record attributes. Columns [risk_ID] and [last_score] are evaluated automatically, so the fields for them were disabled. For each of the other input fields a validation control was implemented to verify that user enters non-empty data for required fields and that the user data input is of the required data type and within the allowed range. The validated fields are checked "on change" and before committing to the database, so it is ensured that no incorrect data type or empty values can be committed and the user input is guaranteed to be secure. The rest of the insert template is represented by two buttons in the bottom of the template – the "Save" and "Cancel" buttons. Button labelled "Save" causes saving filled risk data into database and returning to the standard project risk view, where the newly entered data are already displayed thanks to allowed asynchronous updates of the web part. Button "Cancel" causes cancelling of the insert template and returning to the standard project risk view as well. Implementing Edit and Delete interface for the web part has no sense in this case as the project risks originated once in the project lifecycle are not subject to change and need to be evaluated and kept in mind through entire project life until its finish.

Besides the project leaders, this extended functionality is also available for users belonging to "Site_owners" group. Users with basic access level are not allowed to modify the application content,

so they are only provided the basic version of the Data view. Site administrators are restricted from the access to the Data view at all, because it contains business data not necessary for their scope of work.
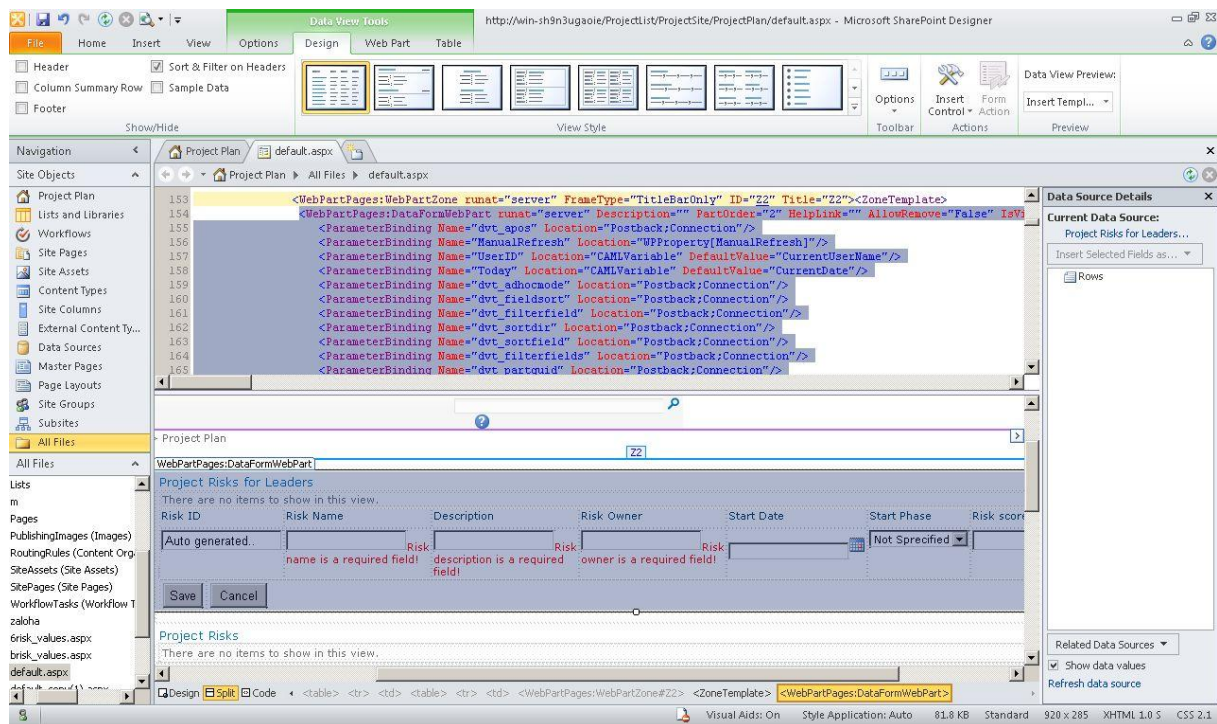
Picture 4. Designing the 'Insert Template' for a new risk.

Site navigation

The site navigation panel was again created by modification of a basic Data view web part without binding it to any data source. The reason for using this web part template was that it allows convenient binding of parameters. The web part was freed of redundant components and unnecessary code. Instead a table with several rows corresponding to application hierarchy levels was inserted. The first row contains link to the upmost application hierarchy level, the project page. Second and third row contain description and a name of the selected project, whose risks are displayed on the current page. The name of a project is coincidentally a link to the current page. Besides, the actual position is always indicated in the navigation panel by underlining of the respective navigation panel element. Last row of navigation panel contains text "Risk history values" to indicate there is one lower level in the application hierarchy. However, this row is not hyperlink- or bold-formatted.

7.5.4   Creating risk history page

This page was also created in application subsite Project Plan. This page represents the last level of application hierarchy displaying the most-detailed information about one selected risk. The risk history values are displayed since risk origin, sorted according the date so the evolution of the risk throughout the project is illustrated.

Risk values history

The data of recorded risk values is stored in the application database, so a Data view web part is used to for the presentation. After Data view was inserted to the page, a new data source had to be created to provide the data from database. The data source was created as a database connection and named "risk_values on Rizika". Columns selected from database were [risk_ID], [date_created], [probability], [consequences] and [score]. The Data view was populated with records for the selected risk, columns were given clear titles and the records were ordered by date of creation to pass consistent information about the risk development. This form of the Data view is accessible for common users.

Inserting new values

Project leaders must be capable of inserting new values, e.g. after risk re-evaluation. Therefore a more advanced version of Data view was created, Insert template was designed and a link for inserting new values added to the bottom of the Data view. In the Insert template the necessary input fields were placed, the Risk ID is pre-filled automatically and the Score field was disabled as the value is computed automatically. Values of Probability and Consequences must be within the range 0-100. The fields were assigned validation controls to validate their type, range and non-empty values before submitting to database. Finally, a "Save" button was added to submit the new record into the database and a "Cancel" button to cancel the insert mode and return to the default Data view. Again, except for "Site_projectLeaders", this version of Data view is accessible also for "Site_owners" user group. "Site_admins" group is restricted from accessing this web part as its content can be a sensitive business information.

Site navigation

Site navigation panel was created in a similar way like on the subsite default page. Parameters containing parent project ID, project name, risk ID and risk name were retrieved from the QueryString variables 'pID', 'pn', 'risk_ID' and 'rn'. Site navigation was realized through hyperlinks pointing to the upper levels of application hierarchy, comprising also the necessary parameters. The hyperlink for the actual page in the form of parent risk name points to the page itself. Hyperlinks were placed into a table with several rows reflecting the application hierarchy and the actual level was marked by underlining the respective hyperlink.

The chart web part

The chart web part was placed to the lower part of the web page. Its purpose is to display risk score evolution in a graphical form. Although it is possible to examine the evolution of risks in time from risk values history, the graphical representation gives a better perspective and provides more meaningful and useful information. The biggest advantage of the chart over the text representation of the risk score values is a better representation of the trend in risk evolution. This trend can be rising, roughly constant or descending, it can be more or less steep and also more or less fluctuating. These are only some of the key qualities of the risk evolution which are in vast majority of cases much more important to the project team than the absolute numerical value of the risk score. They indicate the project and risk health status, correctness of project and risk management and of their settings as well. They also significantly influence risk treatment policy and further project development.

When searching for the suitable web part to graphically present the evolution of risks, several possible candidates were found, the two most promising functionality having the Fusion Charts for

SharePoint beta from FusionCharts Technologies [23] and the Free Google Chart SharePoint Web Part from AMREIN ENGINEERING AG [24]. The third option was using standard Chart web part from the SharePoint web part gallery. The Fusion Charts for SharePoint and the Free Google Chart SharePoint Web Part solutions were downloaded, built in Microsoft Visual Studio 2010 and deployed into the local SharePoint Farm. Their custom properties and functionality were tested to find out which solution fits the best to the developed application needs. In the end, the standard SharePoint Chart web part was chosen to fit the best for the purpose of risk values presentation (Pic. 5).

Picture 5. Risk evolution chart.

Several requirements had to be fulfilled in order to display the risk evolution in a graph. Used Chart web part is connectable to the SharePoint structure as a data source. As the data of risk values are stored in external (non-SharePoint) database, an External Content Type (ECT) had to be defined in SharePoint to make the external data available through an External List structure. The "New External Content Type" button was chosen from the gallery of site External Content Types. After naming the new External Content Type "external_type", the next step was definition of connection to external data source. By clicking on "Add connection" a new connection wizard was launched. Firstly, the type of new connection was selected to be SQL server. Then the server name, database name and a connection method had to be filled. After filling and confirmation of the values, the database 'Rizika' appeared in the Data Source Explorer window.

After navigating to the 'risk_values' table of the database, it was necessary to define External Content Type Operations for our ECT. The set of External Content Type Operations defined for particular External Content Type describes types of operations which SharePoint is capable of when accessing the content of a non-SharePoint data source. After right-clicking the 'risk_values' table in the Data Source Explorer, creating of these operation types were available: New Read Item Operation, New Read List Operation, New Create Operation, New Update Operation, New Delete Operation,

New Association and Create All Operations. Two types of these operations are essential for future creation of external list – Read Item Operation and Read List Operation. Without these basic operations defined over the external content, SharePoint is not able to access external content items and compose a list from them. Therefore these two operations were selected to be created. In the wizard for creating Read Item operations it was necessary to map selected data source columns to identifiers on the Input parameters page and also on the Filter parameters page. Each time at least one data source column had to be mapped, otherwise the external list cannot be created. For the purpose of our list, all columns of 'risk_values' table were mapped as Input and also Filter parameters. Wizard for creating Read List operations allowed selecting filter parameters and mapping them to proper data source element on the first page, so all the 'risk_values' table columns were added and mapped to data source. On the second page of the wizard all table columns were mapped as Return parameters with the responding identifiers. Then the wizard was finished and newly created external type could be saved and closed.

Once the External Content Type was created, it was necessary to set permissions for accessing it. The Central Administration page was open through the web browser and 'Manage Service Applications' selected from the Application Management group. After selecting 'Business Data Connectivity Service', the list of created external content types was displayed. "external_type" was selected and "Set Object Permissions" chosen from the ribbon. Then for 'All authenticated users' and 'All users (windows)' the permission to execute was selected and the permissions window closed by clicking OK button.

After finishing the settings, the newly created External Content Type was marked in the External Content Types gallery of the website in SharePoint Designer and New External List button was selected. The new list was called "external_list". When the list finally existed, it was possible to use it as the data source for the Chart web part.

The data source selection, further settings of the data series and 'Look & Feel' modifications were done through the "Data & Appearance" button of the web part. Another button "Advanced properties" serves for customizing the chart annotations, legends, toolbars, chart areas, titles or context menus. At first, data source for the chart was selected through the "Data & Appearance" – "Connect Chart to Data" menu. In the first step, connecting to a list was selected for retrieving the chart data. Then the list "external_list" was selected from the menu of site's lists. Filtering of list data was skipped in the third step. Finally, the chart was bound to the data in the fourth step. The type of data series displayed in the chart was set to 'Line', as a data source for x-axis was selected the 'date_created' column, for y-axis the 'score' column. Data series were set to be grouped by the 'risk_ref' column. Then the wizard was finished and chart displayed. Next step was customizing the chart look through "Customize Your Chart" wizard of the chart web page. The type of chart was selected to be 'Line without margins'. Then the chart was set to have 0 transparency level on the page and its dimensions were set to 800x600 pixels. In the last step, a title and a legend was added to the chart, axes titles and properties were set and the y-axis was set to have logarithmic scale in order to display the data in more suitable scale for expressing risk severity both in lower and highest levels. Logarithmic scale of the risk score variable allows for more detailed view of the low score values and sufficient overview of the higher values. From this point of view, logarithmic scale is much more convenient and provides more realistic perspective on the risk gravity and its possible influence on the project. Since the values of risk score can range from 0 to 10 000, the logarithmic scale of the y-axis also divides this scope into four implicit levels naturally illustrating degrees of risk seriousness.

7.6 Application migration

The application was developed and is currently running on a testing development server with operating system Windows Server 2008 R2, with software equipment described in previous chapters. This application equipment is rather specific and for the purpose of designing and running the created application it is indispensable. For application running only, not all the software equipment is imperative. The minimal software requirements are 64-bit Windows operating system on which SharePoint can be installed and properly configured, SharePoint Server 2010 platform installed and running, installed SQL database (preferably a version of Microsoft SQL Server) on the server side and web browser MS Internet Explorer 8 or higher on the client side.

The application was developed with the purpose of future experimental using and evaluation in production environment, where it could contribute to a better and more effective management of real IT projects. After application development a way must be specified for migration of the application to another production or testing server and its configuration for correct operation. In general, two such ways are possible to accomplish application repositioning. Either the SharePoint Server site, where the application resides, may be backed up and restored to a different server or the application site can be simply exported to a predefined location and then imported to the SharePoint Farm on another server. The procedures for application export and backup will be described subsequently.

7.6.1 Application site export

Exporting a SharePoint site is one of the ways of migration of a site to different server. Before exporting a SharePoint site, a folder has to be created to store the exported data in. The folder can be created on a local drive or on a network drive or storage. For exporting large sites, however, is more convenient to create to the folder on a local drive due to performance improving. If desired, the folder can be moved to the network after the export is finished. Another precondition for SharePoint site exporting requires the user account, from which the export is performed, to have a "SharePoint_Shell_Access" role for the specified database.

To perform the export itself, a 'SharePoint 2010 Management Shell' from the 'Microsoft SharePoint 2010 Products' group in Start menu has to be started. The export is the performed by submitting following command into the shell:

```
Export-SPWeb -Identity <Site URL> -Path <Path and file name> [-Force]
[-IncludeVersions] [-NoFileCompression] [-GradualDelete] [-Verbose]
```

Parameter `-Identity` precedes the site URL on the server. Parameter `-Path` states the location of the exported site to be saved to. Parameter `-Force` is used to overwrite the specified output file. Different versions can be included into the export with parameter `-IncludeVersions`, which can specify "`All`", "`LastMajor`" (default), "`CurrentVersion`" or "`LastMajorandMinor`" versions of the site to be exported. Parameter `-NoFileCompression` can be used to omit compression while the site is being exported. It can save up to 30% of resources during the site export process. It causes a backup folder with the exported site to be created instead of a compressed file. If the parameter is used while export, it must be used also when the site is imported to its new location. For large site collections, the `-GradualDelete` parameter can be used. It indicates that the exported site collection

will be marked as deleted immediately, but its physical data will be deleted gradually in time, what reduces the one-time impact on the server performance during site export. The `-Verbose` parameter can be used to display the progress of the export operation.

7.6.2   Application backup

Backup of the application is, in comparison to an application export, a more suitable solution. The backup can be performed as an incremental or full, for the selected site collections of whole SharePoint Farm. Moreover, performing backup of the SharePoint site should be a standardly used policy, so no special effort is required to acquire a backup of the selected site collection. Another advantage of using the backup attitude is, that the entire SharePoint Farm can be backed up at once, including its structure, all customizations, user settings and security options by performing a complete full backup. This solution seems to be the most convenient also for this case.

SharePoint Farm or site collection backup can be performed through a SharePoint 2010 Management Shell or through a Central Administration page using the web browser. As backup can affect the performance of SharePoint Farm servers, it is purposeful to perform it when the servers are not much loaded. Before the backup, a folder has to be created to store the backup data in. Again, it can be a location on a local or network drive, but it is recommended to backup the data to local drive and move them thereafter, if necessary, especially when performing a backup of a farm or large sites. Next precondition required is that for creating backup through SharePoint 2010 Management Shell, the user account, under which the backup is performed, must have a "SharePoint_Shell_Access" role for the respective database. When performing the backup through the Central Administration site, the logged on user must have 'farm administrator' rights.

The backup through Central Administration procedure will be described now. The backup was performed for the application site collection first and for the entire SharePoint Farm afterwards.

Site collection backup

In the beginning, Central Administration page was opened in the web browser and "Backup and Restore" group headline selected. Then a "Perform a site collection backup" link was clicked from the "Granular Backup" group. On the next page, the site collection "http://win-sh9n3ugaoie" was selected for the backup, the path to a desired file name was filled and 'Start backup' button clicked. Backup information page showed and after a short while, the backup was finished. The size of bare site collection backup was about 20MB. The backup file is stored on the enclosed DVD in "*/backup/site_backup.bak*".

SharePoint Farm backup

For the entire SharePoint Farm backup, the "Perform a backup" link was selected from the "Farm Backup and Restore" group of the "Backup and Restore" section of Central Administration. Then a page with the farm components was displayed, it was possible to select all by the upper checkbox. On the next page, the options for full or incremental backup, backup scale ant target file location were set. After pressing the "Start backup" button, the farm backup started, displaying the progress and settings information page. The entire SharePoint Farm backup lasted about two hours and the size of output file was approximately 4.5GB. However, after manual compressing the output file into a .zip archive,

the size was significantly decreased and the file was stored on the enclosed DVD in "/backup/farm_backup.zip".

After a farm backup was finished, it was necessary to backup the database which farm uses. For MS SQL Server database it is easy to create a backup using MS SQL Management Studio. After connecting to the database server, the database 'Rizika' was navigated in Object Explorer window. Upon a right-click, 'Tasks' was pointed in the database menu and 'Backup' was selected. Then full backup was chosen and a backup file location specified. Afterwards, the backup was started and completed within minutes. The full database backup is also available on the enclosed DVD in "/backup/db_backup.bak".

7.7   Implementation result

The application was implemented as a SharePoint site solution. The pages were designed using SharePoint Designer 2010, in both Design and Code view. The source code was generated by SharePoint Designer concurrently, based on the operations performed when designing the application site structure and functionality. The extent of four main application pages "ProjectList/default.aspx", "ProjectList/Admin.aspx", "ProjectPlan/default.aspx" and "ProjectPlan/risk_values.aspx" is about 6500 lines of generated XSL code. However, these four files themselves are far from being the entire application solution. The application can be thought of as an entire SharePoint site with defined content, structure, functionality, settings, customizations, templates, back end resources, together with entire farm configuration, and especially operating web services that keep the whole SharePoint architecture moving.

For a closer look, a fragment code example of one of the main Data view web parts is enclosed in Attachment B. The entire source code is stored on the enclosed DVD in "/source/ProjectList/[default.aspx, Admin.aspx]" and "/source/ProjectPlan/[default.aspx, risk_values.aspx]"

8   Fulfilment of requirements

At the beginning of the application design, before implementation itself, an application requirements analysis was worked and a structured set of requirements defined. Fulfilment of these requirements will be discussed in this chapter.

8.1   Requirements assessment

The application requirements specification defined in chapter [6.2] was covering several functional areas and was structured into multiple subsections. They will be evaluated in respective order in next sections, according to their categories.

8.1.1   Functionality

8.1.1.1   Projects

Projects are essential for the application hierarchy. They contain groups of risks that are related to the parent project. Record of each risk influencing a particular project contains an ID of precisely this one parent project. No risk record can exist within the application without having a valid parent project ID. On the contrary, project that has (temporarily) no risks assigned can exist in the application.

8.1.1.2   Risk description

Upon selecting a project on the main application page, all project risks together with their descriptive attributes are displayed. Virtually unlimited number of risks can be added to the projects by project leaders. The list of all recorded risks of a project is displayed in a table after project selection.

8.1.1.3   Main risk attributes

Required attributes of risk probability, consequences and score are defined for each risk. The value of score is automatically evaluated as a product of risk probability and consequences, as this formula is universal enough and provides an overview of risk seriousness. The values of risk probability and consequences can be defined within the range 0-100, risk score can than range from 0 to 10 000. Each risk value record consists of risk ID, date of recording, probability, consequences and score. Risks have also descriptive attributes like name, description, date and phase of the project in which the risk originated, risk owner and risk class depending on their score.

8.1.1.4   Risk evaluation

The value of risk score is calculated according to the formula:
$$score = probability * consequences$$
Risk class is assigned to the risk according to the score value.

8.1.1.5   Re-evaluation

Each risk has its history of values. If the conditions of a project change, project leader can simply record new values of risk attributes into the application. The value of score will be automatically recalculated and risk will be assigned to appropriate class.

8.1.1.6   Old values

Old risk values are saved and can be displayed as a table by selecting the particular risk on the risks application page. The values can be then examined or new values can be inserted.
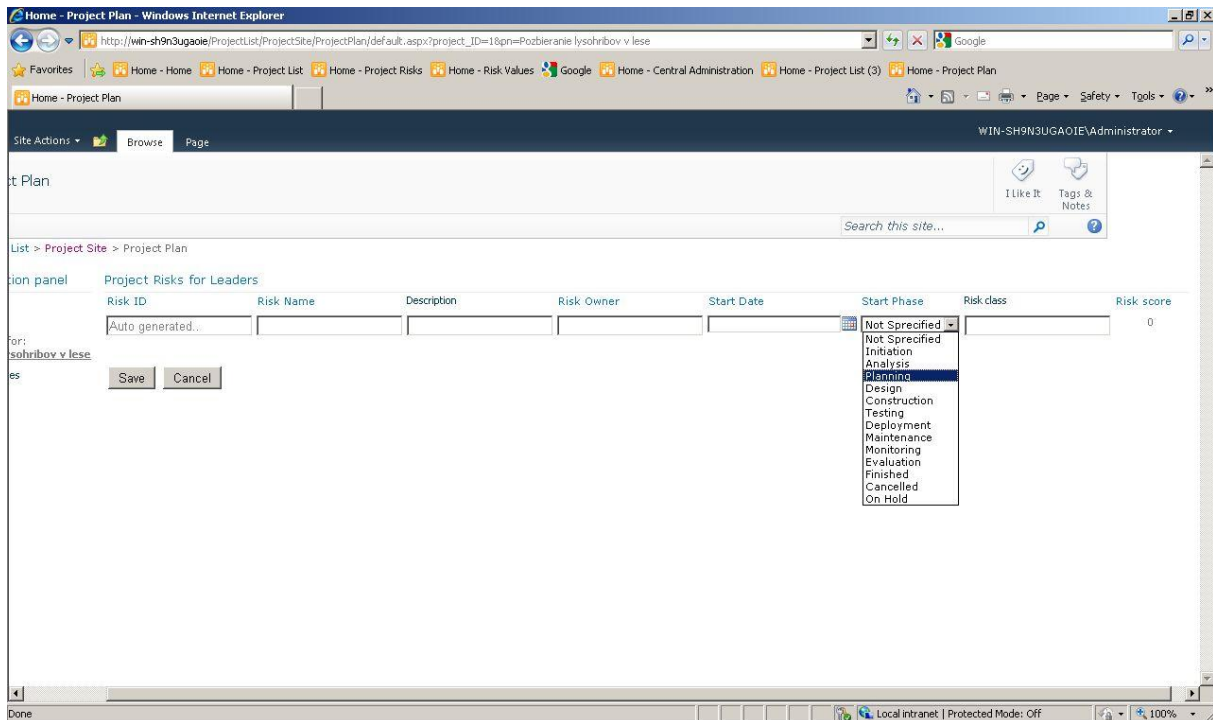
Picture 6. Inserting a new risk record.

8.1.1.7   New risks

Users with sufficient rights can insert new risks into the database and insert their new values afterwards (Pic. 6). The risks and their values can be added in any phase of the project and their values are taken into account when assigning the risk the proper class and evaluating average riskiness of entire project.

8.1.1.8   Evolution in time

The risk values history is saved for each risk in time. Readings from older risk records can be viewed after selecting the particular risk from parent project risks table. As the data are sorted according the time of their recording, the risk evolution can be examined during the life of a risk. Moreover, evolution of risks is also represented graphically by a line chart showing the development of risk values in time.

8.1.1.9   Project stages

System allows setting actual project stage by editing the record of a project at any time. Also each risk has a record of the project stage in which it originated. Risks of a certain project can be also sorted according to the stages of origin. This way the stages of a project can be evaluated from the point of risk occurrence quantity.

8.1.1.10   Printing output

The data from application can be standardly printed on the client's side through a web browser. All of project list page, project risks page and risk history page can be printed, including the chart displaying

the risk evolution. Another way of data external usage can be future interconnecting of application database with other production applications and utilizing the project riskiness information in business. The example of printed output is attached on the enclosed DVD in the root directory under a name printed_output.pdf.

8.1.2 Users and human factor

8.1.2.1 Multi-user application

The application is built on the top of the MS SharePoint 2010 architecture which is primarily designed as multi-user platform and is designated to provide shared content, application and web services to wide user audiences simultaneously. From this point of view, SharePoint is one of the high-end solutions currently available.

8.1.2.2 User differentiation

The levels of user rights and access itself are managed by the application. Users are allowed to access only data of those projects to which they were given access by administrator. After that, administrator has to add user to a specific user group to grant him requested access level. Without being assigned to such group users have no access to the application data. There are three levels of access defined: common user access with read permissions, access for project leaders with read-write permissions and an administrator level access with rights for managing user access and restricted access to application business data.

8.1.2.3 User access

Application was designed to provide user access based on logged-in user account domain name. User has to login only to access the SharePoint Farm. That means that after deployment of the application into a production SharePoint environment users working within this environment are provided proper access level to the application without necessity to be bothered by typing login and password, because the domain account name is used instead (since users are already logged in to the SharePoint Farm). This innovative feature is one of the top application contributions and definitely an advantage to be highly appreciated when used for daily work.

8.1.2.4 Easy-to-use

The application content can be accessed through a common Microsoft Internet Explorer browser, so users should have no problems with its using. The application interface is simple, user-friendly and very intuitive.

8.1.2.5 Editing user profile

This feature was not implemented. It was omitted for several reasons. The main one was that no user profile data is necessary to be stored in the application, not even login or password. The access is managed on a "higher level" through user account domain names which exist outside the application, independently from it. The data about user position is not necessary to be stored either, as the levels of user access are implemented through membership in SharePoint user groups. Management of these

groups, user rights and levels of access is again governed outside the application, through SharePoint Farm users and permissions management features.

8.1.2.6   User management

User management is provided by a SharePoint Farm user management features. As user accounts will be created or removed according to the coming new/leaving current users from the company, every use will already have an account created when he gets to work with the application. After leaving of user from the company, his account will be simply deleted and thus removed also from respective SharePoint user groups defining application access.

8.1.2.7   Authorities

System implements common user level of access with read-only permissions, project leader level access with read-write permissions and an administrator level of access for user access management, with restricted access to business data.

8.1.3   System interface and resources

8.1.3.1   Platform

Application is implemented on a SharePoint platform which makes it highly-available for all authorized users within the company. It can be used by vast number of users simultaneously, without any special software requirements. The area of accessibility can be extended also to company's extranet, after necessary SharePoint Farm configuration.

8.1.3.2   Company resources

The application will be running on a company's SharePoint Farm, so it will be centrally manageable, accessible within intranet and no data need to be sent to outer network. The application can be easily configured by some of the company's SharePoint administrators, if necessary.

8.1.3.3   Simple to launch

Application content is accessible through Internet Explorer which is by default available on each Windows-using computer without any special resources being required. Picture 7 shows the application home page – list of projects available for the logged on user.

8.1.3.4   Simple and reliable inputs

The most authoritative data for projects and their riskiness are probability and consequences values of contained risks. These values are ranged from 0 to 100, to be easily determinable, measured and processed. As risk data are kept under observation for a longer time through the existence of a risk, they should be trustworthy enough to plan further project management on them. Reliability of inserted data is ensured by allowing only project leaders to insert new data.
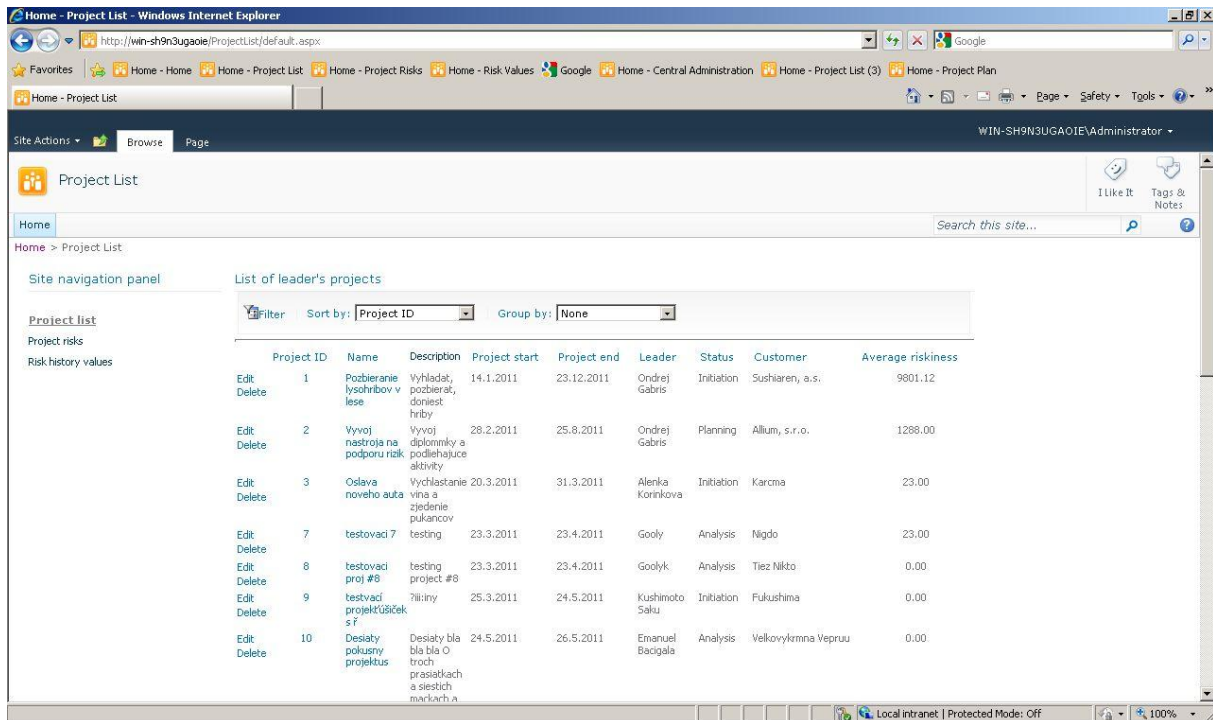
Picture 7. Application Project List page.

8.1.4 Documentation

8.1.4.1 User manual

A User Guide document was composed for describing the functionality of application and scenarios that can be performed by the users having different access levels. This document serves mainly as a reference for new users and as a quick application reference as well. It was named user_guide.pdf and can be found in the root directory of the enclosed DVD.

8.1.4.2 Programming documentation

Description of application development was provided in chapter 7, which can be used as a development guide. The substantial part of the application itself was created using MS SharePoint Designer 2010, in which it can be easily edited. Most features can be customized or upgraded using SharePoint Designer Split view and switching between the code and graphical page design. Some customizations can be done in web browser as described in chapter 7.4.

8.2 Requirements fulfilment summary

The results achieved by designing the application were compared to the requirements specified at the beginning of the development process. These requirements were extracted from practical suggestions and demands after consultations with Ing. Fiedler from Allium, Ltd. Most of the original requirements were met and the business needs were satisfied. Some of the requirements were more closely specified or updated. In one case, the requirement was not implemented, because the requested functionality

was achieved by utilizing the application platform logic and keeping to the original requirement would be more disserviceable than beneficial for the user performance. This way, users were saved from redundant logging into the application, but the security standards for accessing the SharePoint Farm from outside environment were not lowered.

9   Conclusion

The aim of this thesis was elaborating the problem of risks in IT projects, examining thoroughly all stages of the risk management process. Techniques eligible for identification of the risks, their in-depth analysis, assessment, mitigation and prevention were described with respect to the complex risk management process. The Theory of Constraints and Critical Chain Project Management method show that the most important in risk identification, evaluation and management are not complicated formulas and solutions, but the right approach and reliable input data about. The intuitive solutions based on simple methods can be surprisingly efficient and powerful when combined with the right approach and resource utilizing. Demonstration of correct risk management attitude was performed on three sample IT projects. Their risks were analysed and evaluated, consecutively represented on graphs and dividing into separated quadrants was explained. As the classification of risks by a four quadrant graph is not completely common approach, it was used and explained in this thesis to clarify its benefits and higher precision in risk classification, offering more realistic view of the impact and risk position in the project.

The fundamental asset of this thesis is the study of utilizing time-related risk analysis in project risk management and establishing regular re-evaluation of the discovered project risks and searching for the undisclosed ones during entire project life cycle. This was elaborated in detail and demonstrated graphically. The main idea of this attitude in risk management is that the risk conditions of each project are dynamically evolving through the project duration. Therefore the static one-time risk analysis, which is the most common approach in project risk management, is far not sufficient to manage the risk conditions of project during its life period. The risks need to be periodically re-evaluated on reasonable basis and the necessary precautions taken to mitigate arising risks and bring the project to a successful end. Apart from newly originated risks, the existing ones need to be continuously monitored as they can change rapidly in time.

This approach is based on a practical experiences handed over by Mr. Fiedler, who is working as a project leader. Although dynamic risk re-evaluation is not the most common approach in risk management, over time it proves to be a fundamental one for effective project management.

Re-evaluation of risks in time is also supported by the prototype application designed and implemented as a part of this thesis to sustain more effective risk management and handling. It allows project risks overview and evaluation with features for re-evaluation of the risks, saving the history of risk values and representing the risk evolution graphically. Overview of risks in time provides more complex frame of reference on risks evolution and overall project riskiness. From risk evolution point of view, the trend in risk score development is generally more important than the absolute value itself, because it shows the risk behaviour and allows estimating its further tendency. Very useful in this regard is the application graph showing risk trends and long-time evolution. The project management team can then respond differently according to the trend being increasing, decreasing, stable of fluctuating, etc.

The developed application was implemented on Microsoft SharePoint platform, what predestines it for usage in multi-user environment allowing the users to cooperate, share information and collaborate on collective projects. Application also implements a new approach in user security, as each user is given appropriate access without necessity to login to the application itself. The parent

SharePoint Farm is taken as a security authority and provides the user the correct access level. SharePoint platform is also one of the best environments for application future extension and interconnecting with other enterprise systems.

This master thesis was elaborated in a kind cooperation with Mr. Fielder and the Allium, Ltd. company. The company is planning deploying the developed application on their SharePoint Farm to be tested and verified in their production environment to help them in better project risk management. As for an author of the application, it is great to know that the product of this thesis can be used to make some good in a practical life.

Bibliography

1. Pandian, C. Ravindranath: Applied software risk management: a guide for software project managers. Auerbach Publications, 2007, 244 p., ISBN 0-8493-0524-1.

2. Vose, D.: Risk Analysis A quantitative guide. John Wiley & Sons, Inc., 2008, 735 p., ISBN 978-0-470-51284-5.

3. Cooper, D., Grey A., Raymond, G.,Walker, P.: Project risk management guidelines: managing risk in large projects and complex procurements. John Wiley & Sons Ltd, 2005, 384 p., ISBN 0-470-02281-7.

4. Goldratt, E. M.: Critical Chain. The North River Press Publishing Corporation, 1997, 244 p., ISBN 0-88427-153-6.

5. Larssen, L., Lientz B. P.: Risk Management for IT Projects How to Deal with Over 150 Issues and Risks. Elsevier Inc., 2006, 331 p., ISBN 0-7506-8231-0.

6. The Secretary Department of Treasury and Finance: Investment Lifecycle Guidelines Supplementary Guidance #2. Project Risk Management Guideline, 2009.
    Document available from: <www.lifecycleguidance.dtf.vic.gov.au> [2011-01-14]

7. Condamin L., Louisot J., Naim P.: Risk Quantification Management, Diagnosis and Hedging. John Wiley & Sons Ltd, 2006, 271 p., ISBN 0-470-01907-7.

8. Gallati, R. R.: RISK MANAGEMENT AND CAPITAL ADEQUACY. The McGraw-Hill Companies, Inc., 2003, 555 p., ISBN 0-07-140763-4.

9. Mundt A. P.: Dynamic risk management with Markov decision processes, [Dissertation Thesis], Karlsruhe, 2007

10. Chorafas D. N.: Liabilities, Liquidity, and Cash Management Balancing Financial Risks. John Wiley & Sons, Inc., 2002, 316 p., ISBN 0-471-10630-5.

11. Mbokane S.: IT project risk management in the financial sector: an exploratory study, [Short dissertation], Johannesburg, 2005

12. Gabriš O.: Software Projects Risk Management Support Tool, term project, Brno, FIT VUT v Brně, 2011.

13. NOEL, M., SPENCE, C.: *Microsoft SharePoint 2010 Unleashed*. First Printing. United States of America : Pearson Education, Inc., 2010. 939 p., ISBN 978-0-672-33325-5

14. Microsoft: *Microsoft SharePoint 2010* [online]. 2010, 2011 [2011-05-16].
    Document available from: <http://sharepoint.microsoft.com/cs-cz/Pages/default.aspx>.

15. Microsoft: SharePoint 2010 Architectures Overview [online]. 2010 [2011-05-21]
    Document available from: < http://msdn.microsoft.com/en-us/library/gg552610.aspx>.

16. Microsoft: SQL Server 2008 | Prehľad [online]. 2008, 2011 [2011-05-27]. Document available from: < http://www.microsoft.com/slovakia/sqlserver/overview.mspx>.

17. Mistry R.: Microsoft SQL Server 2008 Management and Administration. United States of America: Sams Publishing, 2009. ISBN 978-0-672-33044-5

18. Rizzo T., Alirezaei R., Swider P., Fried J., Hillier S., Schaefer K.: Professional SharePoint® 2010 Development. Indiana: Wiley Publishing, Inc., 2010. ISBN 978-0-470-52942-3

19. Microsoft: Product Information SharePoint Designer 2010 [online]. 2010, 2011 [2011-05-22]. Document available from: < http://sharepoint.microsoft.com/en-us/product/related-technologies/pages/sharepoint-designer.aspx>

20. SharePoint-Videos.com: SharePoint and InfoPath eLearning Video Library [online]. 2011 [2011-03-25]. Document available from: < http://www.sharepoint-videos.com/sp10navigational-components-of-sharepoint-designer-2010>

21. Microsoft: Microsoft® SharePoint® 2010 Professional Developer's Evaluation Guide [online]. 2011 [2011-05-12]. Document available from: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=cffb14e8-88a9-43bd-87aa-4792ab60d320&displaylang=en>

22. SQL Server Helper: Frequently Asked Questions - SQL Server 2005 - VARCHAR(MAX), NVARCHAR(MAX), VARBINARY(MAX) [online]. 2005 [2011-02-31]. Document availble from: < http://www.sql-server-helper.com/sql-server-2005/..%5Cfaq%5Csql-server-2005-varchar-max-p01.aspx>

23. FusionCharts Technologies: Interactive dashboards in SharePoint [online]. 2011 [2011-04-26]. Document available from: < http://www.fusioncharts.com/sharepoint/>

24. AMREIN ENGINEERING AG: Free Google Chart Sharepoint Web Part [online.] 2011 [2011-03-19]. Document available from: <http://www.amrein.com/apps/page.asp?Q=5749>

25. Windischman W., Phillips B., Rehmani A., Kellar M.: Beginning Sharepoint Designer 2010. Indiana: Wiley Publishing, Inc., 2011., ISBN: 978-0-470-64316-7

26. Coventry P.: Microsoft® SharePoint® Designer 2010 Step by Step. Canada: O'Reilly Media, Inc., 2010. ISBN 978-0-735-62733-8

27. Withee K.: SharePoint® 2010 Development For Dummies®. New Jersey: Wiley Publishing, Inc., 2011. ISBN: 978-0-470-88868-1

28. Husman G.: Beginning SharePoint® Administration: Windows® SharePoint Services and SharePoint Portal Server. Indiana: Wiley Publishing, Inc., 2006. ISBN-10: 0-470-03863-2

29. Carter E., Scholl B., Jausovec P.: SharePoint 2010 Development with Visual Studio 2010. United States, Pearson Education, Inc., 2010. ISBN-13: 978-0-321-71831-0

30. Microsoft. *MSDN Magazine* [online]. 2009 [2011-05-15]. Visual Studio 2010 Tools for SharePoint Development. Document available from: <http://msdn.microsoft.com/en-us/magazine/ee309510.aspx>.

31. Microsoft. *SharePoint Developer Center* [online]. 2007 [2011-05-31]. SharePoint Page Viewer Web Part - How to determine the logged in user?. Document available from: <http://social.msdn.microsoft.com/forums/en-US/sharepointdevelopment/thread/acc74f5e-9f28-4a94-b041-27f4d932fa38>.

32. OLIVIER, Johan . *SharePoint Roots* [online]. 2010 [2011-05-31]. SharePoint 2010 Provider Consumer web parts. Document available from: <http://johanolivier.blogspot.com/2010/08/sharepoint-2010-provider-consumer-web.html>.

33. Microsoft. *MSDN Library* [online]. 2010 [2011-05-08]. Connect Web Parts in SharePoint 2010. Document available from: <http://msdn.microsoft.com/en-us/library/ff597538.aspx>.

34. *Fringe SharePoint Continued* [online]. 2011 [2011-05-24]. Web Part Development in SharePoint 2010 - What you need to know! Document available from: <http://geekswithblogs.net/juanlarios/archive/2010/10/24/web-part-development-in-sharepoint-2010---what-you-need.aspx>.

35. *Linglom.com : Source of IT knowledge* [online]. 2009-03-29 [2011-02-27]. Enable Remote Connection on SQL Server 2008 Express. Document available from: <http://www.linglom.com/2009/03/28/enable-remote-connection-on-sql-server-2008-express/>.

36. *Microsoft Office* [online]. 2010 [2011-01-27]. Create data views in SharePoint Designer. Document available from: <http://office.microsoft.com/en-us/sharepoint-designer-help/create-data-views-in-sharepoint-designer-HA101631628.aspx>.

37. HUEN, Stephen . *The Code Project* [online]. 2 May 2010 [2011-03-21]. Event Calendar Listing Web Part (SharePoint 2010). Document available from: <http://www.codeproject.com/KB/sharepoint/SPFCalendarList.aspx>.

38. GOMIUNIK, Boris. *Web BorG* [online]. 2007 [2011-03-04]. QueryString - get parameters from URL with JavaScript. Document available from: <http://webborg.blogspot.com/2007/07/querystring-get-parameters-from-url.html>.

39. K-FÉ, Gustavo . *Blogspot.com* [online]. 2008 [2011-03-02]. Gustavo K-fé Frederico. Document available from: <http://gustavofrederico.blogspot.com/2008/09/passing-parameters-between-pages-via.html>.

40. Microsoft. *SharePoint Developer Center* [online]. 2007 [2011-04-02]. Web Part - Target Audience. Document available from: <http://social.msdn.microsoft.com/forums/en-US/sharepointdevelopment/thread/e18e13e7-a94c-4c23-91d1-45e937622a78/>.

41. GAFFEY, Josh . *Josh Gaffey's Blog* [online]. 2008-06-03 [2011-04-14]. Getting the Current User Name or ID in SharePoint using SharePoint Designer (no code). Document available from: <http://blogs.msdn.com/b/joshuag/archive/2008/06/03/getting-the-current-user-name-or-id-in-sharepoint-using-sharepoint-designer-no-code.aspx>.

42. Microsoft. *MSDN Library* [online]. 2010 [2011-04-25]. How to: Create an External Content Type Based on a SQL Server Table. Document available from: <http://msdn.microsoft.com/en-us/library/ee557243.aspx>.

43. Microsoft. *Microsoft Office* [online]. 2007 [2011-04-25]. Add a database as a data source. Document available from: <http://office.microsoft.com/en-us/sharepoint-designer-help/add-a-database-as-a-data-source-HA010100908.aspx#BM6>.

44. JONES, Jeff . *EndUserSharePoint* [online]. 2010-03-10 [2011-03-26]. SharePoint – [Me] = Easy Item Level Security. Document available from: <http://www.endusersharepoint.com/2010/03/10/sharepoint-me-easy-item-level-security/>.

45. Microsoft. *Microsoft Office* [online]. 2007 [2011-04-30]. Edit, delete, or insert records in a Data View. Document available from: <http://office.microsoft.com/en-us/sharepoint-designer-help/edit-delete-or-insert-records-in-a-data-view-HA010119113.aspx>.

46. *Devguru.com* [online]. 2005 [2011-04-30]. XSLT FUNCTION: format-number . Document available from: <http://www.devguru.com/technologies/xslt/quickref/xslt_function_formatnumber.html>.

47. Microsoft. *SharePoint Developer Center* [online]. 2009 [2011-04-30]. DataView Webpart/Edit Template. Document available from: <http://social.msdn.microsoft.com/Forums/en/sharepointcustomization/thread/38062a56-bb1a-4aba-a7f2-cbe5e054958d>.

48. KARWIN, Bill . *Stackoverflow.com* [online]. 2008 [2011-04-30]. SQL - fetch the row which has the Max value for a column. Document available from: <http://stackoverflow.com/questions/121387/sql-fetch-the-row-which-has-the-max-value-for-a-column>.

49. *Microsoft Technet* [online]. 2010 [2011-05-10]. Plan for Business Connectivity Services (SharePoint Server 2010). Document available from: <http://technet.microsoft.com/en-us/library/ee681491.aspx>.

50. Microsoft. *Microsoft Office* [online]. 2010 [2011-05-10]. Create a chart by using the Chart Web Part. Document available from: <http://office.microsoft.com/en-us/sharepoint-server-help/create-a-chart-by-using-the-chart-web-part-HA101889211.aspx#_Phase_2:_Connect>.

51. *Dotnetspark* [online]. 2010-11-29 [2011-05-01]. Data Form Web Part - Sharepoint Dropdown List as a field in Edit View. Document available from: <http://www.dotnetspark.com/links/59157-data-form-web-part---sharepoint-dropdown.aspx>.

52. RACKLEY, Mark . *EndUserSharePoint* [online]. 2009-11-03 [2011-05-12]. Setting SharePoint Form Fields Using Query String Variables Without Using JavaScript. Document available from: <http://www.endusersharepoint.com/2009/11/03/setting-sharepoint-form-fields-using-query-string-variables-without-using-javascript/>.

53. Microsoft. *MSDN Library* [online]. 2010 [2011-05-20]. CustomValidator.ClientValidationFunction Property . Document available from: <http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.customvalidator.clientvalidationfunction.aspx>.

54. *Stackoverflow.com* [online]. 2010 [2011-05-20]. Validation for SharePoint 2010 external lists / external content types. Document available from: <http://stackoverflow.com/questions/2760774/validation-for-sharepoint-2010-external-lists-external-content-types>.

List of Attachments

Attachment A. Use case descriptions.
Attachment B.   Sample source code of a Data view web part.
Attachment C. SQL query creating [projects_rights_view] view.

Content of enclosed DVD

Folder on DVD	File	Description
root	xgabri06_DP.pdf	This version of Master Thesis.
	printed_output.pdf	Example of site page printed through a web browser.
	user_guide.pdf	User guide for a quick overview of the application functionality.
/backup	db_backup.bak	Full backup of the sample database used to test the application.
	farm_backup.zip	Full backup of the SharePoint farm containing the application site.
	site_backup.bak	Backup of application site.
/source/ProjectList	default.aspx	Home page of Project List site.
	Admin.aspx	Project access administration site.
/source/ProjectPlan	default.aspx	Subsite page for project risks.
	risk_values.aspx	Subsite page for risk values history.
/database	cr_db_rizika.sql	SQL script creating database [Rizika] on the local MS SQL Server.
	cr_projects.sql	SQL script creating table [projects] on the local MS SQL Server.
	cr_rights.sql	SQL script creating table [rights] on the local MS SQL Server.
	cr_risks.sql	SQL script creating table [risks] on the local MS SQL Server.
	cr_risk_values.sql	SQL script creating table [risk_values] on the local MS SQL Server.
	cr_projects_rights_view.sql	SQL script creating view [projects_rights_view] on the local MS SQL Server.

Attachment A

List of all typical cases of application use are described in this attachment. Table representation enables more structured and unified form for characterising the usage and shows sequences of partial operations constituting the whole operation.

1   Use case Add a new risk

ID:	**1**
Label:	**Add new risk**
Created:	Ondrej Gabriš
Description:	Add new risk to the project
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. A project was created by Project leader. Project Leader opens this project.
Postconditions:	New risk is saved in database.
Launched by:	PL chooses "Add new risk".
Main flow:	1) Project Leader chooses "Ad new risk" button. 2) A blank form for adding a risk is displayed. 3) Project Leader fills in the data about new risk. 4) Project Leader confirms entered data. 5) System saves new risk data in database.
Alternative flow:	PL does not confirm the data.
Exceptions:	Running operation failure. Database failure.
Frequency:	Several times a day.
Special conditions:	-

Alternative flow

ID:	**1.1**
Label:	**Add new risk – user does not confirm entered data**
Created:	Ondrej Gabriš
Description:	System does not save entered data, throws them away instead.
Primary actor:	Project Leader
Preconditions:	User chose to cancel entered data instead of saving them in DB.
Postconditions:	Entered data will be thrown away.

Launched by:	Project Leader chooses Cancel in step 4 of use case No. 1
Alternative flow :	1) System throws away data filled into the form.
	2) System returns back to the previous state.
Frequency:	Quite often.

Exceptions

ID:	**1.E.1**
Label:	**Add new risk – Running operation failure**
Created:	Ondrej Gabriš
Description:	Running operation of adding new risk encountered fatal error and will be terminated.
Primary actor:	Project Leader
Preconditions:	System did not carry out the requested operation correctly. Operation failed, but application is still running.
Postconditions:	New risk data is not added to the system.
Launched by:	Application performs an unauthorised or incorrect operation while adding new risk and the operation fails.
Exception flow:	1) After failure, application announces an operation error. 2) Application returns to the previous state before calling the failed operation.
Frequency:	Very rare.

ID:	**1.E.2**
Label:	**Add new risk – Database failure**
Created:	Ondrej Gabriš
Description:	Saving data of new risk into database cannot be completed due to database failure.
Primary actor:	Project Leader
Preconditions:	System could not save new risk data into database. Database failed.
Postconditions:	New risk data is not added into the system.
Launched by:	Database encounters a fatal error or malfunction and is unable to recover from it.
Exception flow:	1) After failure, application announces database error. 2) Application returns to the previous state.
Frequency:	Nearly never.

2 Use case Edit a risk

ID:	**2**
Label:	**Edit a risk**
Created:	Ondrej Gabriš
Description:	Edit risk attributes.
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. A risk in a project was created by Project leader. Afterwards PL opens this risk.
Postconditions:	Modified risk attributes are saved in database.
Launched by:	Project Leader chooses "Edit risk".
Main flow:	1) Project Leader chooses "Edit risk" button. 2) Data of selected risk is displayed in editing mode. 3) PL changes attributes of the risk. 4) PL confirms modified data. 5) System saves modified risk data in database.
Alternative flow:	Project Leader does not confirm the data.
Exceptions:	Running operation failure. Database failure.
Frequency:	Several times a week.
Special conditions:	-

Alternative flow

ID:	**2.1**
Label:	**Edit a risk – user does not confirm modified data**
Created:	Ondrej Gabriš
Description:	System does not save modified data, throws them away instead.
Primary actor:	Project Leader
Preconditions:	User chose to cancel modified data instead of saving them in DB.
Postconditions:	Modified data will be thrown away.
Launched by:	Project Leader chooses Cancel in step 4 of use case No. 2
Alternative flow :	1) System throws away data filled into the form. 2) System returns back to the previous state.
Frequency:	Rarely.

Exceptions

Exceptions in this case are fully analogical to the use case No. 1.

3   Use case Display a risk

ID:	**3**
Label:	**Display a risk**
Created:	Ondrej Gabriš
Description:	Show risk attributes.
Primary actor:	Employee, Project Leader (PL)
Preconditions:	Application is running correctly. A risk was created by Project leader.
Postconditions:	Risk attributes are displayed in a form.
Launched by:	Employee wants to display a risk.
Main flow:	1) Employee chooses the requested risk from the list of project risks. 2) System retrieves data of selected risk from database 3) Data of selected risk is displayed as its attributes' values.
Alternative flow:	-
Exceptions:	Running operation failure. Database failure.
Frequency:	Daily.
Special conditions:	-

Exceptions

ID:	**3.E.1**
Label:	**Display risk – Running operation failure**
Created:	Ondrej Gabriš
Description:	Running operation of displaying risk attributes in form encounters a fatal error and will be terminated.
Primary actor:	Employee/PL
Preconditions:	System did not carry out the requested operation correctly. Operation failed, but application is still running.
Postconditions:	Risk data is not displayed in the prepared form.
Launched by:	Application performs an unauthorised or incorrect operation while processing retrieved risk data and the operation fails.
Exception flow:	1) After failure, application announces an operation error. 2) Application returns to the previous state before calling the failed operation.
Frequency:	Very rare.

ID:	**3.E.2**
Label:	**Display risk – Database failure**
Created:	Ondrej Gabriš
Description:	Retrieving data of a risk from database cannot be accomplished due to database failure.
Primary actor:	Employee/PL
Preconditions:	System could not retrieve risk data from database. Database failed.
Postconditions:	Risk data is not read into the system.
Launched by:	Database encounters a fatal error /malfunction and is unable to recover from it.
Exception flow:	1) After the failure, application announces database error. 2) Application returns to the previous state.
Frequency:	Nearly never.

4   Use case Evaluate/Re-evaluate project risks

ID:	**4**
Label:	**Evaluate/Re-evaluate project risks**
Created:	Ondrej Gabriš
Description:	Evaluate risks of a project according to their relevant attributes.
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. A project was created by Project leader. Risks were added to the project and their attributes are set. The project is open.
Postconditions:	Risks are evaluated and desiderative values are calculated
Launched by:	Employee/PL chooses to display project risks.
Main flow:	1) Employee chooses to display project risks. 2) Attributes of respective risks are retrieved from database. 3) Additional risk attribute values are calculated, based on retrieved data. 4) Newly calculated attribute values are presented. 5) Newly calculated data are saved into database if necessary.
Alternative flow:	-
Exceptions:	Running operation failure. Database failure.

Frequency:	Several times a day.
Special conditions:	-

Exceptions

Exceptions in this case are fully analogical to the use case No. 1.

5 Use case Display risk evolution graph

ID:	**7**
Label:	**Display risk evolution graph for the project risks**
Created:	Ondrej Gabriš
Description:	Shows graphical representation of risks evolution in time.
Primary actor:	Employee, Project Leader (PL)
Preconditions:	Application is running correctly. A project was created by Project leader. Risks were added to the project and their attributes are set. The project is open and risks have been evaluated.
Postconditions:	Graphical time-lined representation of risk evolution is plotted. Graph can be saved to an external format.
Launched by:	Employee displays the project risk information.
Main flow:	1) Employee displays the project risk information. 2) Attributes of respective risks are retrieved from database. 3) Evolution of risks is plotted on graph.
Alternative flow:	-
Exceptions:	Plotting operation failure. Database failure.
Frequency:	Several times a week.
Special conditions:	-

Exceptions

Exceptions in this case are fully analogical to the use case No. 1.

6 Use case Print risk evolution graph

ID:	**8**
Label:	**Print risk evolution graph**
Created:	Ondrej Gabriš

Description:	Prints graphical representation of risks evolution.
Primary actor:	Employee, Project Leader (PL)
Preconditions:	Application is running correctly. A project was created by Project leader. Risks were added to the project and their attributes are set. The project is open and risks have been evaluated.
Postconditions:	Graphical time-lined representation of risk evolution is printed.
Launched by:	Employee chooses "Print preview" from the "File" context menu of the browser.
Main flow:	1) Employee chooses "Print preview" from the "File" context menu of the browser. 2) User sets the requested printing parameters of requested format and numbers of pages to print. 3) Evolution graph of risks is sent to a printer.
Alternative flow:	-
Exceptions:	Printer failure/loss of connection. Plotting operation failure. Database failure.
Frequency:	Several times a week.
Special conditions:	-

Exceptions

Exceptions in this case are fully analogical to the use case No. 6.

7 Use case Set up a new project

ID:	**10**
Label:	**Set up a new project**
Created:	Ondrej Gabriš
Description:	Establish a new project in application.
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. User has the rights of project leader.
Postconditions:	New project is saved in database.
Launched by:	PL chooses "Add new project".
Main flow:	1) PL chooses "Add new project" button. 2) A blank form for setting up new project attributes is displayed. 3) Project leader fills in the data about new project. 4) Project leader confirms entered data.

	5) System saves newly established project in database.
Alternative flow:	PL does not confirm the data.
Exceptions:	Running operation failure.
	Database failure.
Frequency:	Not very often.
Special conditions:	-

Alternative flow

ID:	**10.1**
Label:	**Set up a new project – user does not confirm entered data**
Created:	Ondrej Gabriš
Description:	System does not save newly filled project data, throws them away instead.
Primary actor:	Project Leader
Preconditions:	User chose to cancel entered project data instead of saving them in DB.
Postconditions:	Entered data will be thrown away, project is not created.
Launched by:	PL chooses Cancel in step 4 of use case No. 10
Alternative flow :	1) System throws away data filled into the form.
	2) System returns back to the previous state.
Frequency:	Rarely.

Exceptions

ID:	**10.E.1**
Label:	**Set up a new project – Running operation failure**
Created:	Ondrej Gabriš
Description:	Running operation of setting new project encountered fatal error and will be terminated.
Primary actor:	Project leader
Preconditions:	System did not carry out the requested operation correctly.
	Operation failed, but application is still running.
Postconditions:	New project data is not added to the system.
Launched by:	Application performs an unauthorised or incorrect operation while adding newly established project information and the operation fails.
Exception flow:	1) After failure, application announces an operation error.
	2) Application returns to the previous state before calling the failed operation.

Frequency:	Very rare.

ID:	**10.E.2**
Label:	**Set up a new project – Database failure**
Created:	Ondrej Gabriš
Description:	Saving data of newly set project into database cannot be accomplished due to database failure.
Primary actor:	Project leader
Preconditions:	System could not save new project data into database. Database encountered unrecoverable problem and failed.
Postconditions:	New project data is not saved into the system. Database is down.
Launched by:	Database encounters a fatal error or malfunction and is unable to recover from it.
Exception flow:	1) After failure, application announces database error. 2) Application returns to the previous state.
Frequency:	Nearly never.

8 Use case Delete selected project

ID:	**11**
Label:	**Delete selected project**
Created:	Ondrej Gabriš
Description:	Delete a project from system.
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. User has the rights of project leader. Project, that is to be deleted, was previously created in system.
Postconditions:	Project is no longer in database/system.
Launched by:	PL chooses "Delete project".
Main flow:	1) PL chooses "Delete project" button. 2) System allows PL to select project to be deleted. 3) Project leader chooses the project. 4) Project leader clicks on link for deleting the project. 5) System erases the project record and all risks it contained from database. Also all rights records to access the projects are erased.
Alternative flow:	PL does not perform the deletion.
Exceptions:	Running operation failure. Database failure.

Frequency:	Not very often.
Special conditions:	-

Alternative flow

ID:	**11.1**
Label:	**Delete selected project – user does not perform deletion**
Created:	Ondrej Gabriš
Description:	System does not delete the project record and contained risks from database.
Primary actor:	Project Leader
Preconditions:	PL did not perform the deletion of chosen project.
Postconditions:	Selected project and its risks will be kept in database.
Launched by:	PL chooses does not click the 'Delete' link in step 4 of use case No. 11
Alternative flow :	1) System does not delete the project. 2) System remains in the previous state.
Frequency:	Rarely.

Exceptions

Exceptions are fully analogical to those in use case No. 10.

9   Use case Edit project properties

ID:	**12**
Label:	**Edit project properties**
Created:	Ondrej Gabriš
Description:	Edit properties of existing project and save them in database.
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. User has the rights of project leader. Selected project was previously created and exists in system.
Postconditions:	Edited project properties are saved in database.
Launched by:	PL chooses "Edit project".
Main flow:	1) PL chooses "Edit project" button. 2) Properties of selected project are loaded from database. 3) A form filled with project properties is displayed in editable mode. 4) Project leader edits required properties. 5) Project leader confirms edited data. 6) System saves project properties in database.

Alternative flow:	PL does not confirm the data.
Exceptions:	Running operation failure.
	Database failure.
Frequency:	Not very often.
Special conditions:	-

Alternative flow

ID:	**12.1**
Label:	**Edit project properties – user does not confirm entered data**
Created:	Ondrej Gabriš
Description:	System does not save edited project data project record in system remains unchanged.
Primary actor:	Project Leader
Preconditions:	User chose to cancel edited project data instead of saving them in DB.
Postconditions:	Entered data will be thrown away, project record is not modified.
Launched by:	PL chooses Cancel in step 5 of use case No. 12
Alternative flow :	1) System throws away data filled into the form.
	2) System returns back to the previous state.
Frequency:	Rarely.

Exceptions

Exceptions are fully analogical to those in use case No. 10.

10 Use case Create a new user access

ID:	**13**
Label:	**Create new user access**
Created:	Ondrej Gabriš
Description:	Create new user access for selected project
Primary actor:	Administrator
Preconditions:	Application is running correctly. User has administrator rights.
Postconditions:	New user with defined level of user rights is recorded in system.
Launched by:	Administrator chooses "Add new user".
Main flow:	1) Administrator chooses "Add new user" button.
	2) A blank form for filling user domain name is displayed.
	3) Administrator fills in the data and confirms.
	4) He also adds user to a respective user group according to the level of

	user rights.
	5) System saves new user in database.
Alternative flow:	Administrator does not confirm creation of user.
Exceptions:	Running operation failure.
	Database failure.
Frequency:	Rarely.
Special conditions:	-

Alternative flow

ID:	**13.1**
Label:	**Create new user – administrator does not confirm creation of user**
Created:	Ondrej Gabriš
Description:	System does not create new user.
Primary actor:	Administrator
Preconditions:	Administrator did not confirm saving new user in system.
Postconditions:	Entered data will be thrown away.
Launched by:	Administrator chooses Cancel in step 3 of use case No. 13
Alternative flow :	1) System throws away data filled into the form.
	2) System returns back to the previous state.
Frequency:	Very rarely.

Exceptions

ID:	**13.E.1**
Label:	**Create new user access – Running operation failure**
Created:	Ondrej Gabriš
Description:	Running operation of creating new user access encountered fatal error and will be terminated.
Primary actor:	Administrator
Preconditions:	System did not carry out the requested operation correctly.
	Operation failed, but application is still running.
Postconditions:	New user data is not added to the system, it is lost instead.
Launched by:	Application performs an unauthorised or incorrect operation while creating new user and the operation fails.
Exception flow:	1) After failure, application announces an operation error.
	2) Application returns to the previous state before calling the failed operation.
Frequency:	Very rare.

ID:	**13.E.2**
Label:	**Create new user access – Database failure**
Created:	Ondrej Gabriš
Description:	Saving data of newly created user access into database cannot be completed due to database failure.
Primary actor:	Administrator
Preconditions:	System could not save new user data into database. Database failed.
Postconditions:	New user data is not added into the system. Database is down.
Launched by:	Database encounters a fatal error or malfunction and is unable to recover from it.
Exception flow:	1) After failure, application announces database error. 2) Application returns to the previous state.
Frequency:	Nearly never.

11   Use case Delete selected user's access

ID:	**15**
Label:	**Delete selected user's access**
Created:	Ondrej Gabriš
Description:	Delete chosen user's domain name from system.
Primary actor:	Administrator
Preconditions:	Application is running correctly. Deleting performing user has the rights of administrator. User domain name that is to be deleted exists in system.
Postconditions:	User is no longer in database/system.
Launched by:	Administrator chooses "Remove this user".
Main flow:	1) Administrator selects the project for which user has access. 2) Administrator has the list of users with access to project displayed. 3) Administrator chooses the user. 4) Administrator chooses "Remove this user" button 5) System erases the user record from database.
Alternative flow:	Administrator does not perform the deletion.
Exceptions:	Running operation failure. Database failure.

Frequency:	Not very often.
Special conditions:	-

Alternative flow

ID:	**15.1**
Label:	**Delete selected user – administrator does not perform deletion**
Created:	Ondrej Gabriš
Description:	System does not delete user record from database.
Primary actor:	Administrator
Preconditions:	Administrator did not perform the deletion of chosen user, he cancelled it.
Postconditions:	Selected user will be kept in database.
Launched by:	Administrator chooses not to delete the user in step 4 of use case No. 15
Alternative flow :	1) System does not delete the user. 2) System returns back to the previous state.
Frequency:	Rarely.

Exceptions

Exceptions are fully analogical to those in use case No. 14.

12   Use case Open project

ID:	**17**
Label:	**Open project**
Created:	Ondrej Gabriš
Description:	Display selected project information & attributes.
Primary actor:	Employee, Project Leader (PL)
Preconditions:	Application is running correctly. A project was created and exists in system.
Postconditions:	Project data and attributes are displayed in a form.
Launched by:	Employee chooses to open selected project by clicking on it.
Main flow:	1) Employee clicks on a name or ID of selected project. 2) System retrieves data of selected project from database. 3) Data of selected project is displayed as its attributes' values.
Alternative flow:	-
Exceptions:	Running operation failure.

	Database failure.
Frequency:	Daily.
Special conditions:	-

Exceptions

ID:	**17.E.1**
Label:	**Open project – Running operation failure**
Created:	Ondrej Gabriš
Description:	Running operation of opening a project encounters a fatal error and will be terminated.
Primary actor:	Employee/PL
Preconditions:	System did not carry out the requested operation correctly. Operation failed, but application is still running.
Postconditions:	Project data is not displayed in the prepared form.
Launched by:	Application performs an unauthorised or incorrect operation while processing retrieved risk data and the operation fails.
Exception flow:	1) After failure, application announces an operation error. 2) Application returns to the previous state before calling the failed operation.
Frequency:	Very rare.

ID:	**17.E.2**
Label:	**Open project – Database failure**
Created:	Ondrej Gabriš
Description:	Retrieving data of a project from database cannot be accomplished due to database failure.
Primary actor:	Employee/PL
Preconditions:	System could not retrieve project data from database. Database failed.
Postconditions:	Project data is not read into the system. Database is down.
Launched by:	Database encounters a fatal error /malfunction and is unable to recover from it.
Exception flow:	1) After the failure, application announces database error. 2) Application returns to the previous state.
Frequency:	Nearly never.

13   Use case Set project phase

ID:	**18**
Label:	**Set project phase**
Created:	Ondrej Gabriš
Description:	Set project phase and save it.
Primary actor:	Project Leader (PL)
Preconditions:	Application is running correctly. Selected project was previously created and exists in system.
Postconditions:	Set project phase is saved in database.
Launched by:	User chooses "Edit project".
Main flow:	1) User chooses "Edt project" button. 2) Properties of respective project are loaded from database. 3) User sets the project phase. 4) User confirms edited data. 5) System saves project properties in database.
Alternative flow:	User does not confirm the data.
Exceptions:	Running operation failure. Database failure.
Frequency:	Not very often.
Special conditions:	-

Alternative flow

ID:	**18.1**
Label:	**Set project phase – user does not confirm entered data**
Created:	Ondrej Gabriš
Description:	System does not save edited project data, project record in system remains unchanged.
Primary actor:	Project Leader
Preconditions:	User chose to cancel edited project data instead of saving them in DB.
Postconditions:	Entered data will be thrown away, project record is not modified.
Launched by:	PL chooses Cancel in step 4 of use case No. 18
Alternative flow :	1) System throws away data filled into the form. 2) System returns back to the previous state.
Frequency:	Rarely.

Exceptions

Exceptions are fully analogical to those in use case No. 10.

Attachment B

This is a fragment of central Data view web part from page "ProjectList/default.aspx". As the entire code is very extensive, the original page is attached on the enclosed DVD in location "/source/ProjectList/default.aspx"

```
<WebPartPages:WebPartZone runat="server" FrameType="TitleBarOnly" ID="Z2"
Title="Z2"><ZoneTemplate>
                           <WebPartPages:DataFormWebPart runat="server" Description=""
ListDisplayName="" PartOrder="2" HelpLink="" AllowRemove="False" IsVisible="True"
AllowHide="False" UseSQLDataSourcePaging="True" ExportControlledProperties="True"
DataSourceID="" Title="List of leader's projects" ViewFlag="8" NoDefaultStyle="TRUE"
AllowConnect="True" FrameState="Normal" PageSize="-1" PartImageLarge="" AsyncRefresh="True"
ExportMode="All" Dir="Default" DetailLink=""
AuthorizationFilter=";;;;Site_projectLeaders,Site_owners" ShowWithSampleData="False"
ListId="00000000-0000-0000-0000-000000000000" AllowClose="False" ListName=""
FrameType="TitleBarOnly" PartImageSmall="" IsIncluded="True" SuppressWebPartChrome="False"
AllowEdit="False" ManualRefresh="False" ChromeType="TitleOnly" AutoRefresh="False"
AutoRefreshInterval="60" AllowMinimize="False" ViewContentTypeId=""
InitialAsyncDataFetch="False" MissingAssembly="Cannot import this Web Part."
HelpMode="Modeless" ListUrl="" ID="g_d93266db_4899_4aa4_9790_552aeae29e4b"
ConnectionID="00000000-0000-0000-0000-000000000000" AllowZoneChange="False"
IsIncludedFilter=";;;;Site_projectLeaders,Site_owners" __MarkupType="vsattributemarkup"
__WebPartId="{D93266DB-4899-4AA4-9790-552AEAE29E4B}" __AllowXSLTEditing="true" WebPart="true"
Height="" Width=""><ParameterBindings>
                           <ParameterBinding Name="dvt_apos"
Location="Postback;Connection"/>
                           <ParameterBinding Name="ManualRefresh"
Location="WPProperty[ManualRefresh]"/>
                           <ParameterBinding Name="UserID" Location="CAMLVariable"
DefaultValue="CurrentUserName"/>
                           <ParameterBinding Name="Today" Location="CAMLVariable"
DefaultValue="CurrentDate"/>
                           <ParameterBinding Name="dvt_1_form_editkey"
Location="Postback;Connection"/>
                           <ParameterBinding Name="dvt_1_form_insertmode"
Location="Postback;Connection"/>
                           <ParameterBinding Name="name" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="description" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="project_start" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="project_end" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="leader" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="status_finished" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="current_phase" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="customer" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="domain_name" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="right_level" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="original_project_ID" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="original_name" Location="None"
DefaultValue=""/>
                           <ParameterBinding Name="original_description" Location="None"
DefaultValue=""/>
```

```xml
                                                <ParameterBinding Name="original_project_start"
Location="None" DefaultValue=""/>
                                                <ParameterBinding Name="original_project_end" Location="None"
DefaultValue=""/>
                                                <ParameterBinding Name="original_leader" Location="None"
DefaultValue=""/>
                                                <ParameterBinding Name="original_status_finished"
Location="None" DefaultValue=""/>
                                                <ParameterBinding Name="original_current_phase"
Location="None" DefaultValue=""/>
                                                <ParameterBinding Name="original_customer" Location="None"
DefaultValue=""/>
                                                <ParameterBinding Name="original_domain_name" Location="None"
DefaultValue=""/>
                                                <ParameterBinding Name="original_right_level" Location="None"
DefaultValue=""/>
                                                <ParameterBinding Name="project_ID" Location="None"
DefaultValue=""/>
                                                <ParameterBinding Name="dvt_adhocmode"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_filterfield"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_sortdir"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_sortfield"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_adhocfiltermode"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_filtertype"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_filterval"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_groupdir"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_groupdisplay"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_groupfield"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_groupsorttype"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_grouptype"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_sorttype"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_fieldsort"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_filterfields"
Location="Postback;Connection"/>
                                                <ParameterBinding Name="dvt_partguid"
Location="Postback;Connection"/>
                            </ParameterBindings>
<DataFields>@project_ID,project_ID;@name,name;@description,description;@project_start,project_
start;@project_end,project_end;@leader,leader;@status_finished,status_finished;@current_phase,
current_phase;@customer,customer;@domain_name,domain_name;@project_ref,project_ref;@w_avg,w_av
g;</DataFields>
<Xsl>
<xsl:stylesheet xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" version="1.0" exclude-result-prefixes="xsl msxsl ddwrt"
xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime"
xmlns:asp="http://schemas.microsoft.com/ASPNET/20"
xmlns:__designer="http://schemas.microsoft.com/WebParts/v2/DataView/designer"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt"
xmlns:SharePoint="Microsoft.SharePoint.WebControls" xmlns:ddwrt2="urn:frontpage:internal">
        <xsl:output method="html" indent="no"/>
        <xsl:param name="dvt_adhocmode">sort</xsl:param>
        <xsl:decimal-format NaN=""/>
        <xsl:param name="UserID">CurrentUserName</xsl:param>
        <xsl:param name="dvt_apos">&apos;</xsl:param>
        <xsl:param name="ManualRefresh"></xsl:param>
        <xsl:param name="dvt_1_form_editkey" />
```

```
        <xsl:param name="dvt_1_form_insertmode" />
        <xsl:param name="dvt_filterfield" />
        <xsl:param name="dvt_sortdir">ascending</xsl:param>
        <xsl:param name="dvt_sortfield" />
        <xsl:param name="dvt_adhocfiltermode">xsl</xsl:param>
        <xsl:param name="dvt_filtertype" />
        <xsl:param name="dvt_filterval" />
        <xsl:param name="dvt_groupdir">ascending</xsl:param>
        <xsl:param name="dvt_groupdisplay" />
        <xsl:param name="dvt_groupfield" />
        <xsl:param name="dvt_groupsorttype">text</xsl:param>
        <xsl:param name="dvt_grouptype" />
        <xsl:param name="dvt_sorttype">text</xsl:param>
        <xsl:param name="dvt_fieldsort" />
        <xsl:param name="dvt_filterfields" />
        <xsl:param name="dvt_partguid" />
        <xsl:variable name="dvt_1_automode">1</xsl:variable>
        <xsl:template match="/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
xmlns:asp="http://schemas.microsoft.com/ASPNET/20"
xmlns:__designer="http://schemas.microsoft.com/WebParts/v2/DataView/designer"
xmlns:SharePoint="Microsoft.SharePoint.WebControls">
                <xsl:choose>
                        <xsl:when test="($ManualRefresh = 'True')">
                                <table width="100%" border="0" cellpadding="0" cellspacing="0">
                                        <tr>
                                                <td valign="top">
                                                        <xsl:call-template name="dvt_1"/>
                                                </td>
                                                <td width="1%" class="ms-vb" valign="top">
                                                        <img
src="/_layouts/images/staticrefresh.gif" id="ManualRefresh" border="0" onclick="javascript:
{ddwrt:GenFireServerEvent('__cancel')}" alt="Click here to refresh the dataview."/>
                                                </td>
                                        </tr>
                                </table>
                        </xsl:when>
                        <xsl:otherwise>
                                <xsl:call-template name="dvt_1"/>
                        </xsl:otherwise>
                </xsl:choose>
        </xsl:template>
        <xsl:template name="dvt_1">
                <xsl:variable name="dvt_StyleName">Table</xsl:variable>
                <xsl:variable name="Rows" select="/dsQueryResponse/NewDataSet/Row[@domain_name
= $UserID]"/>
                <xsl:variable name="dvt_FieldNameNoAtSign" select="substring-
after($dvt_filterfield, '@')" />
                <xsl:variable name="dvt_FilteredRowsText" select="$Rows[.=$dvt_filterval or
($dvt_filtertype='date' and substring-before($dvt_filterval,'T') = substring-before(.,'T'))]"
/>
                <xsl:variable name="dvt_FilteredRows" select="$Rows[normalize-
space(*[name()=$dvt_filterfield])$dvt_filterval or ($dvt_filtertype='date' and substring-
before($dvt_filterval,'T') = substring-before(normalize-
space(*[name()=$dvt_filterfield]),'T'))]" />
                <xsl:variable name="dvt_FilteredRowsAttr" select="$Rows[normalize-
space(@*[name()=$dvt_FieldNameNoAtSign])=$dvt_filterval or ($dvt_filtertype='date' and
substring-before($dvt_filterval,'T') = substring-before(normalize-
space(@*[name()=$dvt_FieldNameNoAtSign]),'T'))]" />
                <xsl:variable name="dvt_RowCount">
                <xsl:choose>
                        <xsl:when test="$dvt_adhocfiltermode != 'query' and $dvt_filterfield">
                                <xsl:choose>
                                        <xsl:when test="starts-with($dvt_filterfield,
'@')"><xsl:value-of select="count($dvt_FilteredRowsAttr)" /></xsl:when>
                                        <xsl:when test="$dvt_filterfield = '.'"><xsl:value-of
select="count($dvt_FilteredRowsText)" /></xsl:when>
                                        <xsl:otherwise><xsl:value-of
select="count($dvt_FilteredRows)" /></xsl:otherwise>
                                </xsl:choose>
```

```
                </xsl:when>
                <xsl:otherwise><xsl:value-of select="count($Rows)" /></xsl:otherwise>
        </xsl:choose>
        </xsl:variable>
        <xsl:variable name="IsEmpty" select="$dvt_RowCount = 0" />
        <xsl:variable name="dvt_IsEmpty" select="$dvt_RowCount = 0"/>
        <xsl:call-template name="dvt_1.toolbar">
                <xsl:with-param name="Rows" select="$Rows" />
        </xsl:call-template>
        <xsl:choose>
                <xsl:when test="$dvt_IsEmpty">
                        <xsl:call-template name="dvt_1.empty"/>
                        <table border="0" width="100%" cellpadding="2" cellspacing="0">
                                <tr valign="top">
                                        <xsl:if test="$dvt_1_automode = '1'"
ddwrt:cf_ignore="1">
                                                <th class="ms-vh" width="1%"
nowrap="nowrap"></th>
                                        </xsl:if>
                                        <th class="ms-vh" nowrap="nowrap" valign="top"
style="text-align: center">
                                                <xsl:call-template name="dvt.headerfield"
ddwrt:atomic="1" xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime">
                                                        <xsl:with-param
name="fieldname">@project_ID</xsl:with-param>
                                                        <xsl:with-param name="fieldtitle">Project
ID</xsl:with-param>
                                                        <xsl:with-param
name="displayname">project_ID</xsl:with-param>
                                                        <xsl:with-param
name="sortable">1</xsl:with-param>
                                                        <xsl:with-param
name="fieldtype">x:long</xsl:with-param>
                                                </xsl:call-template>

                                        </th>
                                        <th class="ms-vh" nowrap="nowrap" style="width:
300px">
                                                <xsl:call-template name="dvt.headerfield"
ddwrt:atomic="1" xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime">
                                                        <xsl:with-param
name="fieldname">@name</xsl:with-param>
                                                        <xsl:with-param
name="fieldtitle">Name</xsl:with-param>
                                                        <xsl:with-param
name="displayname">name</xsl:with-param>
                                                        <xsl:with-param
name="sortable">1</xsl:with-param>
                                                        <xsl:with-param
name="fieldtype">x:string</xsl:with-param>
                                                </xsl:call-template>
                                        </th>
                                        <th class="ms-vh" nowrap="nowrap" style="width:
300px">
                                                <xsl:call-template name="dvt.headerfield"
ddwrt:atomic="1" xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime">
                                                        <xsl:with-param
name="fieldname">@description</xsl:with-param>
                                                        <xsl:with-param
name="fieldtitle">Description</xsl:with-param>
                                                        <xsl:with-param
name="displayname">description</xsl:with-param>
                                                        <xsl:with-param
name="sortable">0</xsl:with-param>
                                                        <xsl:with-param
name="fieldtype">x:string</xsl:with-param>
                                                </xsl:call-template>

                                        </th>
```

Attachment C

The SQL query designing the projects_rights_view looks as follows:

```sql
CREATE VIEW [dbo].[projects_rights_view]
AS
SELECT     dbo.projects.project_ID, dbo.projects.name,
dbo.projects.description, dbo.projects.project_start,
dbo.projects.project_end, dbo.projects.leader,
                   dbo.projects.status_finished,
dbo.projects.current_phase, dbo.projects.customer, dbo.rights.id,
dbo.rights.right_level, dbo.rights.domain_name,
                   CASE WHEN SUM(dbo.risks.last_consequences) IS NULL OR
                   SUM(dbo.risks.last_consequences) = 0 THEN 0 ELSE
SUM(dbo.risks.last_consequences * dbo.risks.last_score) /
SUM(dbo.risks.last_consequences) END AS w_avg,
                   dbo.rights.project_ref
FROM          dbo.projects INNER JOIN
                   dbo.rights ON dbo.projects.project_ID =
dbo.rights.project_ref LEFT OUTER JOIN
                   dbo.risks ON dbo.projects.project_ID =
dbo.risks.project_ref
GROUP BY dbo.projects.project_ID, dbo.projects.name,
dbo.projects.description, dbo.projects.project_start,
dbo.projects.project_end, dbo.projects.leader,
                   dbo.projects.status_finished,
dbo.projects.current_phase, dbo.projects.customer, dbo.projects.average,
dbo.rights.id, dbo.rights.right_level, dbo.rights.domain_name,
                   dbo.rights.project_ref

GO
```