

# Význam manažmentu rizík pre úspešnosť projektu

BC. DANIELA MILOŇOVÁ

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
danka\_mi [zavináč] centrum [.] sk*

**Abstrakt.** V každom projekte sa skôr alebo neskôr vyskytnú riziká, ktoré sa môžu odraziť na kvalite alebo finančnej stránke projektu. Manažment rizík je stále viac uznávaný, zaoberá sa pozitívnymi a negatívnymi stránkami rizika a predstavuje proces, počas ktorého sa riziko identifikuje, analyzuje a vyberie optimálny spôsob pre jeho zvládnutie pri minimálnych potrebných nákladoch. Cieľom manažmentu rizík je minimalizovať dopad vzniknutých rizík alebo v niektorých prípadoch im dokonca aj predísť. V tejto práci sa zameriam na priblíženie práce manažmentu rizík. Sústredím sa na spôsoby identifikovania a analýzy rizík ako aj ich riadenie. Porovnam jednotlivé metódy a v závere zhodnotím ich použitie v rámci nášho tímového projektu.

## Úvod

Vytvorenie dobrého a zákazníkom akceptovateľného softvérového produktu nie je takým častým javom akoby sa na prvý pohľad zdalo. Veľa projektov síce skončí odovzdaním produktu zákazníkovi, ale ich dokončenie je na úkor časového plánu a dodržania plánovaných zdrojov. Toto konanie má za následok nielen stratu zisku a možností, ale aj sklamanie zákazníka. Pri spätnom skúmaní týchto nedostatkov väčšina prípadov poukazuje na to, že ich neúspech mohol byť spôsobený tým, že neboli pripravení na možné nepriaznivé udalosti alebo boli použité zlé opatrenia proti nim. Jedná sa hlavne o problémy s nedodržaním termínu prípadne prekročením finančných zdrojov.

Predpokladať, že sa v projekte nevyskytnú žiadne udalosti, ktoré by mohli spôsobiť stratu alebo poškodenie projektu, je vysoko nereálne, a preto bezpodmienečnou súčasťou každého projektu by mal byť manažment rizík.

## Riziká v softvérových projektoch a ich manažment

Všeobecne riziko môžeme chápať ako pravdepodobnosť vzniku nežiadúcej a nezamýšľanej udalosti a negatívneho dôsledku tejto udalosti. Čiže ak hovoríme o riziku, hovoríme ešte len o pravdepodobnosti, že môže nastať stav ohrozenia. Riziko správa:

- neistota: udalosť, s ktorou je dané riziko spojené, môže, ale nemusí nastať
- strata: následkom sú nečakané dôsledky

Manažment rizík predstavuje proces, počas ktorého sa riziko identifikuje, analyzuje a vyberie optimálny spôsob pre jeho zvládnutie pri minimálnych potrebných nákladoch. Skladá sa z nasledujúcich častí [1]:

- Identifikácia rizík
- Analýza rizík
- Plánovanie rizík
- Riadenie rizík

### Identifikácia rizík

Je prvou fázou manažmentu rizika a slúži na zistenie, ktoré riziká môžu ovplyvniť projekt. Tvorba zoznamu možných rizík vychádza z faktov, že čo všetko sa môže pokaziť behom riešenia projektu a ktoré udalosti k tomu vedú. V súčasnosti existuje mnoho techník a metód pre identifikáciu rizík, takými najbežnejšími sú [5]:

- Audity a inšpekcie
- Diskutovanie a takzvaný brainstorming
- Rozhodovacie stromy
- Posudok experta
- Interview so zákazníkom
- Analýza scenárov
- Skúsenosti a znalosti firmy, organizácie alebo jedincov
- Dotazníky

Identifikácia rizika je správna ak poskytuje úplnosť a máme spôsob kontrolovania tejto činnosti. Podstatnými časťami identifikácie sú presne určené pravidlá a dokumentovanie každého kroku. Dokumenty obsahujú opis rizika, podmienky jeho výskytu, dopad na projekt, ako často a ako pravdepodobne sa môže vyskytnúť (frekvencia), stanovenie priority rizika a spôsob riešenia krízovej situácie, ako sa vyhnúť riziku alebo ho zmierniť.

### **Analýza rizík**

Slúži na zoradenie rizík podľa ich dôsledkov na ciele projektu. Kvantitatívna analýza rizík slúži naopak na určenie pravdepodobnosti rizík a na prepojenie týchto rizík na ciele projektu. Podľa [4] analýza rizík je komplex opatrení zahŕňajúci nasledujúce činnosti:

- Stanovenie hraníc – treba zvážiť, čo všetko sa bude analyzovať a v akom rozsahu
- Identifikácia a ocenenie – treba vytvoriť zoznam všetkých dôležitých častí softvéru a vykonať ich ohodnotenie na základe napr. ceny alebo časovej dostupnosti a pod. Toto ocenenie musí zohľadniť účel a priority v rámci celého softvéru.
- Identifikácia a ocenenie hrozieb – treba zistiť aké hrozby pôsobia na softvér a aká je pravdepodobnosť ich výskytu.
- Identifikácia a ocenenie slabín (zraniteľnosť) – treba identifikovať slabé miesta softvéru, oceniť ich a rozhodnúť o ich eliminácii.
- Profil a miera rizika – zahŕňa ocenenie dopadu jednotlivých hrozieb na konkrétne miesta softvéru. Vo všeobecnosti možno tento proces nazvať stanovením miery ohrozenia.
- Návrh proti opatrení – hlavným cieľom týchto opatrení je eliminovať dopad hrozieb na konkrétne časti.

### *Metódy analýzy rizík*

1. **Metóda ALE (MALE)** – vhodná pre hodnotenie rizík aj vo veľkých systémoch v čase, keď sú už prevádzkované. Vtedy už sú dostatočne známe všetky hmotné aktíva, ktoré systém zahŕňa a možno aj s odpovedajúcou presnosťou stanoviť dopad hrozieb z hľadiska zničenia alebo narušenia integrity dát, straty dostupnosti a pod.
2. **Metóda matíc (MM)** – je výhodná pre systémy lokálneho charakteru či už vo fáze vývoja alebo využívania. Jej výhodou je schopnosť podchytiť veľké množstvo vzájomných vzťahov medzi hmotnými a nehmotnými aktívami, hrozbami a slabinami. Vyžaduje však kvalifikovaný personál, ktorý musí byť schopný veľmi dobre zhodnotiť funkčné a dátové aspekty informačného systému.
3. **Metóda anketovej analýzy (MAA)** – využiteľná pre všetky skupiny systémov v čase prevádzky. Kladením vhodných otázok možno získať množstvo kvalitných informácií využiteľných v metóde matíc (pri stanovení hodnôt prvkov matice) a v metóde podpory rozhodovania (pri distribúcii váhových koeficientov v uzloch rozhodovacieho stromu).

4. **Metóda analýzy scenárov (MAS)** – sú opodstatnené vo veľkých systémoch v čase ich projekcie aj využitia. Základom týchto metód sú analytické alebo dokonca simulačné modely, a preto musia byť vytvorené na mieru pre každú aplikáciu. Vyžadujú vysoko kvalifikovaný analytický tím a zodpovedajúce náklady.
5. **Metódy využitia skúseností (MVS)** – majú len obmedzený význam a možno ich aplikovať len pre niektoré menšie lokálne systémy. Sú založené na podobnosti s iným systémom, pre ktorý už existujú preverené rizikové charakteristiky. Čím je systém väčší, tým ťažšie je nájsť vhodnú analógiu.
6. **Metóda DELPHI panelu (MDP)** – predpokladá existenciu vysoko kvalifikovaného kolektívu, ktorého každý člen je schopný hodnotiť hrozby, slabiny a potenciálne straty. Táto metóda je výhodná najmä pri riešení krízových situácií v rozsiahlych systémoch, ktoré sú už v prevádzke. Z jednej strany vyžaduje riešenie krízovej situácie vysokú odbornosť, na druhej strane však existuje relatívne obmedzený priestor (vymedzený „súradnicami“ HROZBA, DOPAD, OBNOVENIE PÔVODNÉHO STAVU), v ktorom možno metódou konsenzu dospieť k optimálnemu riešeniu.

### Plánovanie

Plánovanie slúži na vytvorenie techník a procedúr na posilnenie pozitívnych rizík a zníženie negatívnych rizík. Plánovanie rizík vlastne definuje činnosti, ktoré sa majú vykonať pre elimináciu, redukciu alebo akceptovanie rizika. Definujú sa rôzne stratégie a vždy sa vyberie najvhodnejšia z nich. Často pri plánovaní nesmieme zabudnúť, že riziká vplývajú vzájomne na seba, čo sa zohľadňuje pri vykonaní opravných a eliminačných akcií.

### Riadenie rizík

Posledná, ale nemenej dôležitá, fáza manažmentu rizík sa zaoberá monitorovaním dlhodobých rizík, identifikovaním nových rizík, vykonávaním plánov na znižovanie rizík a vyhodnocovaním ich efektov počas celého vývoja projektu. Cieľom je čo najskôr zbadáť nepriaznivé situácie, ktoré sú spôsobené vynechaním rizík zo zoznamu identifikovaných rizík, alebo nepočítaním s takou formou nastatia rizika. Riadenie rizika potom zabezpečuje opätovné spustenie cyklu manažmentu rizík: identifikácia, analýza, plánovanie, riadenie. Riadenie rizík je teda opravný proces.

### Riziká

Na zvládnutie rizík treba okrem teoretických znalostí aj skúseností a hlavne odhad. V svojich článkoch sa Dr. Barry W. Boehm [2] podelil so svojimi skúsenosťami vytvoril súhrn desiatich najčastejšie sa vyskytujúcich rizík. V jeho zozname sa v dodržanom poradí nachádzajú:

1. Deficit ľudských zdrojov

2. Nereálny harmonogram a rozpočet
3. Vývoj zlých funkcií a vlastností
4. Vývoj zlého používateľského rozhrania
5. Pozlátenie systému – pridávanie viac do systému ako bolo požadované
6. Prúd pravidelných požiadaviek na zmenu
7. Deficit externe dodávaných komponentov
8. Deficit externe vykonávaných úloh
9. Slabá výkonnosť
10. Vysoké zaťažovanie odborníkov

Podľa inej štúdie [3] boli rizikové faktory rozdelené do 4 kvadrantov (pozri Obr. 1). Vrchná a spodná polovica podľa ich relatívnej dôležitosti a na pravú a ľavú polovicu podľa miery kontroly rizika. V rámci týchto kvadrantov uvediem roztriedené Boehmove riziká.

Miera rizika	vysoká	1 Záujem pozorovateľa	2 Rozsah a požiadavky
	stredná	4 Prostredie	3 Vykonávanie
		nízka	vysoká
		Miera kontroly	

Obr. 1. Kategorizácia rizík [3]

### ***Kvadrant 1. Zapojenie používateľa.***

V tomto kvadrante sa nachádzajú viaceré z najdôležitejších rizík. Pre riešenie týchto rizík sa ukázalo podstatné udržiavať dobré vzťahy so zákazníkom a zvyšovať jeho záujem o spoluprácu pri jeho vývoji. Presvedčiť zákazníka aby prejavoval viac záujmu ako len prvotné zadanie projektu je náročné. Treba sa hlavne zamerať na jeho očakávania. Podľa vlastných skúseností klient nevie čo presne chce, ale približne má predstavu čo si od toho sľubuje.

Riešenie ako si s týmito rizikami poradiť:

- zaistenie, že užívatelia a nie programátori smerujú vývoj. Preto je dôležité upovedomiť budúcich používateľov o ich dôležitej roli pri vývoji.
- patri sem:

- 3. Vývoj zlých funkcií a vlastností
- 4 Vývoj zlého používateľského rozhrania

#### *Kvadrant 2. Rozsah a požiadavky.*

Rizikovým faktorom v tomto kvadrante je nesprávne určenie rozsahu a funkcionality softvérového produktu. Určiť rozsah a požiadavky okamžite na začiatku projektu je takmer nemožné a preto je bežným javom, že v priebehu projektu táto úloha nadobúda konkrétnejšie obrisy. Pre tento kvadrant a riziká ním spojené sú vhodné evolučné prístupy. Najčastejším problémom je nepochopenie alebo aj nestálosť klientských požiadaviek, ktoré sa môžu meniť aj v priebehu procesu.

- dohodnutá funkcionality je rozdelená na nutnú časť, bez ktorej výsledný produkt nevyhovuje. Ďalšou je doplnková časť, ktorá síce nie je bezpodmienečne nevyhnutná pre jeho fungovanie, ale svojím prínosom produkt rozšíri a obohatí
- špecifikovanie ktoré požiadavky sú ešte akceptovateľné a ktoré sú už žiaľ nespĺniteľné, pretože sa zvyšuje zložitosť systému, náklady na jeho vývoj a môžu dokonca ohroziť aj dodržanie časového plánu a tým aj oneskoriť čas dodania
- patrí sem:
  - 5. Pozlátenie systému
  - 6. Prúd pravidelných požiadaviek na zmenu

V rámci riešenia je niekoľko možností. Štruktúrovaná metodológia vhodná, keď sú dopredu jasné požadované vlastnosti produktu, zatiaľ čo napr. rýchle prototypovanie uľahčuje identifikovanie požiadaviek užívateľa. Podľa mňa, keďže zákazník nemá ucelenú predstavu je vhodné používať evolučné prístupy k vývoju softvéru, špirálový model, prototypovanie, agilné metódy a pod.

#### **Kvadrant 3. Vykonávanie.**

Do tohto kvadrantu spadajú rizikové faktory týkajúce sa nedostatočného alebo nepripraveného personálu, výber neefektívnej metodológie, nesprávna definícia rolí a povinností. Zahŕňa všetky riziká spojené so samotným vývojom systému.

- rozdelenie projektu na menšie a lepšie prehľadné časti
- patrí sem:
  - 1. Deficit ľudských zdrojov
  - 2. Nereálny harmonogram a rozpočet
  - 9. Slabá výkonnosť
  - 10. Vysoké zaťažovanie odborníkov

#### *Kvadrant 4. Prostredie*

Posledný kvadrant zahŕňa riziká z prostredia projektu vo vnútri i mimo organizácie. Typickými zástupcami z tejto oblasti sú zmeny na manažérskych postoch a konflikty medzi oddeleniami, prípadne s dodávateľmi. Tieto riziká patria medzi najťažšie riaditeľné a nevyskytujú sa často. Patrí sem:

- 7. Deficit externe dodávaných komponentov
- 8. Deficit externe vykonávaných úloh

### **Záver**

Výsledky manažmentu rizík nie sú navonok viditeľné, znamená to však, že sme sa úspešne vyhli rizikovým situáciám a teda náš manažment rizík funguje správne..

Pri písaní tejto práce som získala prehľad o možných rizikách, s ktorými sa môžeme stretnúť pri vývoji veľkých softvérových projektoch, ale je možné ich aplikovať aj na náš tímový projekt. V rámci nášho projektu sa venujem aj plánovaniu a som si vedomá že chybné naplánovanie môže ohroziť plynulý vývoj projektu. Dobré časové odhady podľa mňa vyžadujú určité skúsenosti z minulých podobných projektov. Najjednoduchším riešením je nechať si v rozvrhu dostatočnú časovú rezervu. Nie je to však vždy možné, preto je potrebné si stanoviť si priority a využívať nejaký podporný prostriedok na plánovanie.

Presné špecifikovanie požiadaviek tu hrá vysokú rolu. Prvotná komunikácia so zadávateľom znamená určenie, čo sa vlastne chystáme riešiť. Je však vhodné ho informovať a zapájať pri vývoji riešenia, aby sme sa vyhli nechcenému odkloneniu od želaného výsledku.

### **Použitá literatúra**

1. Bieliková, M.: Softvérové inžinierstvo. Princípy a manažment. Vydavateľstvo STU, Bratislava 2000.
2. Boehm, B.W.: Software Risk Management: Principles and Practices, 1991
3. Keil, M., Cule, P.E., Lyytinen, K., Schmidt, R.C.: A framework for identifying software project risks. Communications of the ACM, Vol. 41, No. 11 (1998), 76–83.
4. Krbilová, I., Nagy, P., Peniak, P.: Informačné systémy, Žilinská univerzita v Žilina, 1998

### **Annotation**

#### *Importance of risk management for project success*

We can find risks in some part of almost all projects. Risks can bring a new problems or show us a new solutions. That is the reason, why is risk management still more and more important in

IT. Risk management search for a positive and negative parts of risk. Risk management show us how to identify risks, how to analyse it and find optimal solutions. Objective of a risk management is to minimalism problems of existing risks or prevent them. In my essay I will try to show you how it works. I will show how to identify and analyse risks and how to solve them. I will also compare different methods of solutions. In last part of my essay I will apply all my studies to my team project.