

Prečo vôbec analyzovať a plánovať riziká?

ONDREJ POK

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
ondrej.p@zoznam.sk*

Abstrakt. Otázka, ktorú si určite položilo už mnoho ľudí, ktorí sa bez predchádzajúcej prípravy pod vplyvom rôznych okolností dostali do roly manažéra v softvérovom projekte. V tejto eseji sa pokúsim zamyslieť nad oboma stranami mince, teda výhodami aj nevýhodami manažmentu rizík a ako manažment rizík ovplyvňuje náklady na projekt a úspešnosť projektu. Pokúsim sa zvážiť veľkosť a rozsah projektu, pretože projekty malého rozsahu môžu byť ohrozené inými hrozbami ako projekty veľkého rozsahu, ako aj rozdielne metódy vývoja softvéru použité v softvérových projektoch, pretože v súčasnosti stále populárnejšie agilné metódy vývoja si vyžadujú odlišný prístup manažmentu rizík ako plánovo-orientované metódy.

Neraz sa stane, že človek ktorý s riadením projektu nemá nijaké skúsenosti sa dostane do pozície manažéra projektu. Či už kvôli neochote vedenia firmy zabezpečiť odborníka, nedostatku kvalifikovaných odborníkov alebo preto, že jeho schopnosti považujú za dostatočné na túto funkciu.

Jeho schopnosti riadenia tímu nechajme bokom. Pre nás je zaujímavé jeho rozhodnutie ako sa postaviť k rizikám, ktoré môžu ohroziť priebeh a výsledky projektu. Aké výhody pre jeho projekt bude mať, ak sa rozhodne pre manažment rizík? Čo sa môže stať ak bude riziká jednoducho ignorovať?

Ak sa rozhodne pre manažment rizík, je veľká šanca že sa projekt skončí úspešne. Ak sa s manažmentom rizík ešte nestretol, pravdepodobne bude riziká ignorovať, a riešiť ich až *ad hoc*, teda v čase keď dôjde k problémom.

Vasa

Ako odstrašujúci príklad môžem uviesť projekt, ktorý síce nie je z oblasti vývoja softvéru, ale z inžinierskej disciplíny stavby lodí. V článku [1] autori analyzovali dôvody zlyhania projektu stavby vojnovéj lode Vasa Švédskeho námorníctva a hľadali

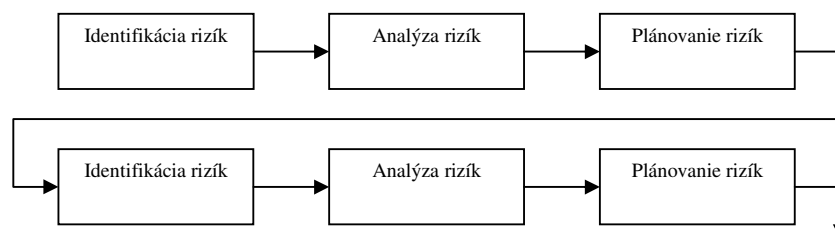
analógie k softvérovým projektom. Dôvodov zlyhania tohto projektu bolo mnoho. Príliš veľký tlak na skoré dokončenie, neustále zmeny požiadaviek od zadávateľa, nedostatočná dokumentácia, príliš veľa inovácií, s ktorými nikto nemal skúsenosti, nedostatok vedeckých metód, ignorovanie jasných varovných signálov, ale najmä nedostatok komunikácie medzi zadávateľom, výročom a používateľmi. Pri stavbe tejto lode došlo k toľkým prešľapom a závažným pochybeniam, že loď sa potopila niekoľko minút po tom, ako bola spustená na vodu.

Aby softvérové projekty nestihol podobný osud ako spomínanú loď, je nanajvýš žiadúce zaviesť manažment rizík. To ale nie je vždy jednoduché. Podľa [2] sú najčastejšie príčiny, pre ktoré sa vo firme nedarí zaviesť manažment rizík, nedostatočné povedomie a celkové vyhýbanie sa tejto otázke, nevhodná organizačná štruktúra firmy, a nedostatok podpory od vedenia.

Manažment rizík

Manažment rizík sa skladá z troch činností: identifikácia rizík, analýza rizík a plánovanie rizík [6]. Identifikácia rizík zahŕňa vytvorenie zoznamu hrozieb, ktoré by mohli projekt ovplyvniť. V kroku analýzy sa určia pravdepodobnosti ich výskytu a možné dopady na priebeh a výsledky projektu. V kroku plánovania rizík sa rozhodne, proti ktorým hrozbám sa vykonajú opatrenia na detekciu alebo ich elimináciu, a naplánujú sa konkrétne opatrenia. Potom je potrebné opatrenia už len implementovať.

Tieto tri kroky by sa mali počas celého projektu cyklicky opakovať, pretože počas práce na projekte sa môžu objaviť ďalšie hrozby, alebo niektoré, s ktorými sa od začiatku počítalo sa stanú zanedbateľnými, napríklad preto, že boli spojené s konkrétnou fázou projektu, ktorá ale už bola úspešne ukončená.



Obr.1 Manažment rizík ako opakujúci sa proces

Ako vidíme, na manažment rizík je potrebný čas, a ľudia, ktorí ho budú vykonávať. To niekedy v projektoch s nízkym rozpočtom, pevnými termínmi a obmedzeným počtom ľudí nie je jednoduché vyčleniť. Ale je možné k manažmentu rizík pristupovať rôznym spôsobom. Podľa [5] sú okrem možnosti vôbec neanalyzovať riziká ešte ďalšie 4 prístupy podľa hĺbky analýzy rizík:

1. Použiť **základný prístup** pre riešenie všetkých rizík – aplikujú sa základné opatrenia pre všetky identifikované riziká, bez ohľadu na ich pravdepodobnosť výskytu alebo možné dopady
2. Použiť na analýzu rizík **neformálny prístup** a sústrediť sa na identifikované riziká s najvyššou pravdepodobnosťou výskytu alebo s najhoršími dopadmi
3. Vykonať detailnú analýzu rizík s využitím **formálneho prístupu**
4. Vykonať počiatočnú analýzu rizík „na vysokej úrovni“ s cieľom identifikovať najväčšie a najkritickejšie riziká. Potom aplikovať opatrenia pre tieto kritické riziká a základné opatrenia pre všetky ostatné identifikované riziká.

Najdôkladnejšia analýza rizík sa dosiahne použitím tretej možnosti, ale je to zároveň časovo a finančne najnáročnejší spôsob analýzy, ktorý môže projekt značne oneskoriť. Preto sa v praxi často používajú aj ostatné prístupy.

Z prehľadu je vidieť, že analýza rizík sa dá robiť aj lacnejším a jednoduchším neformálnym prístupom, pričom aj ten poskytuje vyššiu šancu na úspešné ukončenie projektu ako úplné ignorovanie rizík.

Napríklad podľa [3] je vhodné sledovať a plánovať len „*Top 10*“, čiže 10 najzávažnejších rizík, ktoré môžu ohroziť úspešné ukončenie projektu a nestrácať čas s málo pravdepodobnými rizikami, alebo s takými, ktoré majú len minimálne dopady. Ďalej uvádzam najčastejšie riziká, ktoré hrozia pri softvérových projektoch. Sú to:

- **problémy s personálom** (*nedostatok alebo odchod ľudí alebo nedostatočná kvalifikácia*),
- **nerealistické odhady** (*plány a rozpočty*),
- **nedostatočná komunikácia so zadávateľom a používateľmi** (*vyvíjanie zbytočných funkcionalít a nezrozumiteľných používateľských rozhraní*),
- **neustále zmeny požiadaviek**,
- **nekompatibilita medzi modulmi vyvíjanými rôznymi tímami**,
- **nedostatočná výkonnosť aplikácie v reálnom prostredí**.

Všetky tieto riziká je ale možné minimalizovať vhodnými opatreniami.

Problémy s odchodom personálu sa dajú riešiť dostatočnou motiváciou zamestnancov, alebo takým pridelovaním úloh, aby vždy minimálne dvaja pracovali na úlohe a vedeli sa navzájom zastúpiť. Nedostatočná kvalifikácia sa dá riešiť školeniami, alebo zabezpečením konzultanta.

Na odhady sa dajú použiť expertné nástroje. Aby sa neprekročil rozpočet, je možné prispôsobiť návrh dostupným zdrojom (*design to cost*). Podobne je možné zabezpečiť dodržanie plánov inkrementálnym vývojom.

Aby sa nevyvíjali zbytočné funkcionality, ktoré zákazník ani nepotrebuje, je potrebná úzka spolupráca s používateľmi počas celého vývoja softvéru. Vhodné je aj rýchle prototypovanie, a skorá dodávka používateľských príručiek, ktoré môžu včas odhaliť nepochopenie požiadaviek ešte skôr ako bolo čokoľvek implementované. Je vhodné aby návrhári poznali procesy v spoločnosti pre ktorú je softvér vyvíjaný (*business process modelling*).

Zmeny v špecifikácii je potrebné zapracovať pre spokojnosť zákazníka, ale pri použití inkrementálneho vývoja je možné požiadavky na zmeny zapracovať až v neskorších fázach.

Nekompatibilita medzi modulmi vyvíjanými rôznymi tímami, alebo časťami tímu sa dá riešiť dôkladnou špecifikáciou alebo rýchlym prototypovaním a skorým testovaním rozhrania.

Ako vidieť, na každú hrozbu, ktorá by mohla ohroziť projekt sa dá nájsť účinné protiopatrenie. Na účinné protiopatrenia je ale najprv potrebné poznať hrozby, ktoré projekt ohrozujú.

Výhody (a nevýhody?)

Manažment rizík so sebou prináša veľa výhod a len málo nevýhod. Začnem teda nevýhodami.

Na manažment rizík je potrebné vyčleniť časť rozpočtu projektu, alebo navýšiť rozpočet o túto položku. Je najvhodnejšie poveriť projektového manažéra aby zabezpečil manažment rizík, alebo je potrebné najat' experta v danej oblasti. Okrem finančnej stránky, manažment rizík odčerpáva aj zdroje v podobe času. S týmto faktom teda treba počítať aj pri odhadovaní dĺžky trvania projektu, ako aj pri tvorbe plánov.

Podľa môjho názoru výhody, ktoré manažment rizík prináša, vysoko prevyšujú nad nevýhodami. Najväčším prínosom je ale odhalenie možných rizík ešte predtým, ako spôsobia problém. Riešenie problémov, ktoré už nastali si skoro vždy vyžaduje oveľa viac úsilia a prostriedkov ako prevencia. Riešenie problému navyše oneskoruje projekt, pretože maximum zdrojov je často presunutých na riešenie problému.

Ostatné výhody sú už len dôsledkami včasného odhalenia možných rizík. Vyhnutie sa problémom umožní dokončenie projektu podľa plánu, alebo s menším oneskorením ako keby riziká neboli včas odhalené, a softvér dodaný načas zabezpečí aj spokojnosť zadávateľa.

V dlhodobom horizonte zabezpečí manažment rizík nižšie náklady spojené s riešením problémov, pretože umožní vyhnúť sa väčšine problémov.

Rozsah projektu, metódy vývoja

Je samozrejmé, že rozsah projektu bude mať vplyv aj na to, aké riziká prichádzajú do úvahy, alebo do akej miery sú ktoré riziká závažné.

Problémy s personálom sa budú dotýkať skôr menších projektov ako veľkých projektov na ktorých pracujú desiatky až stovky zamestnancov. Pri takom veľkom počte je jednotliviec jednoduchšie nahraditeľný. To ale nemusí platiť pre každú pozíciu, a zvlášť pre vysoko špecializovaných odborníkov bez ohľadu na rozsah projektu.

Nerealistické odhady sa týkajú rovnako malých ako aj veľkých projektov. Vo veľkých projektoch je ale každé oneskorenie omnoho drahšie, ak zoberieme do úvahy náklady na všetkých pracovníkov, ktorí sa na projekte podieľajú.

Problémy s nekompatibilitou modulov sa vyskytujú častejšie pri projektoch veľkého rozsahu, ktorý sa skladá v viacerých mohutných modulov na ktorých pracujú samostatné tímy. Celková integrácia týchto modulov môže priniesť nečakané problémy.

Ostatné zo zoznamu najčastejších rizík podľa [3] sa týkajú rovnako veľkých projektov, ako aj malých projektov.

Zaujímavým aspektom, ktorý ovplyvňuje aj prístup k manažmentu rizík, je paradigma vývoja softvéru [4].

Plánovo-orientovaný prístup sa najčastejšie využíva vo veľkých projektoch. Najväčšou hrozbou pre takéto projekty sú časté zmeny požiadaviek zo strany zadávateľa. Každá zmena spôsobí prepracovanie veľkého množstva dokumentov, a nové pridelenie úloh.

Na druhej strane stoja agilné metódy vývoja. Využívajú sa hlavne v malých projektoch, do ktorých sú zapojené maximálne desiatky pracovníkov, ktorí medzi sebou úzko spolupracujú. Z toho vyplývajú vážne problémy v prípade výpadku alebo odchodu niektorého zo zamestnancov.

Prečo teda analyzovať a plánovať riziká?

Dôvodov prečo by sa projektový manažér mal rozhodnúť pre dôsledný manažment rizík je mnoho. Najdôležitejším dôvodom je ale to, že s pomocou manažmentu rizík sa možné problémy môžu odhaliť už v predstihu a s pomocou vhodných protipatrení sa týmto problémom dá úplne vyhnúť. Manažment rizík je potrebné vykonávať počas celého projektu, pretože hrozby ohrozujúce projekt sa neustále menia. Je potrebné identifikovať všetky potenciálne riziká, ale najväčšiu pozornosť zamerať len na tie, ktoré môžu vážne ovplyvniť úspešnosť projektu a aplikovať proti nim opatrenia.

Konečným dôsledkom úspešného manažmentu rizík je menej problémov počas vývoja pre vývojový tím, spokojnosť zadávateľa aj používateľov a dobré meno dodávateľskej firmy.

Použitá literatúra

1. Fairley, Richard & Willshire, Mary Jane: *Why the Vasa sank – 10 problems and some antidotes for Software projects*, IEEE Software, March/April 2003, pp. 18-25
2. Carr, Marvin: *Risk management may not be for everyone*, IEEE Software, May/June 1997, 21-24
3. Boehm, Barry: *Software risk management: principles and practices*, IEEE Software, January 1991, 32-41
4. Boehm, Barry & Turner, Richard: *Using risk to ballance agile and plan-driven methods*, Computer, June 2003, 57-66
5. Strnád, Ondrej: *Manažment bezpečnosti IT*, Slovenská technická univerzita v Bratislave, 2002

6. Bieliková, Mária : *Manažment v softvérovom inžinierstve*, Bratislava, 1999

Annotation

Why analyse and plan risks?

A question many people, who had been assigned the role of a project manager without previous experience, asked themselves.

In this essay I will mention both pros and cons of risk management and how the risk management influences the costs and the project success. I will try to consider also the project scope, because large projects can be threatened by different threats than small projects. I will also consider the the paradigm of the software development, because the stil more popular agile development methods are threatened by different threats from for instance plan-driven development methods.