

Manažment rizík v softvérovom projekte

MIROSLAV LAŠŠÚ

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
lassu.miroslav[zavináč]gmail[.]com*

Abstrakt. Riziko, hlavne také, ktoré môže spôsobiť úplný koniec projektu, je niečo, na čo si treba dávať poriadny pozor, ale aj napriek tomu je zlyhanie softvérového projektu až prekvapivo bežná záležitosť. Vo väčšine týchto prípadov by sa zlyhaniu dalo zabrániť dodržiavaním určitých zásad a metód, ktoré sa označujú ako manažment rizík. Manažment rizík zahŕňa postupy ako riziká identifikovať, analyzovať a odstrániť, prípadne minimalizovať ich výskyt a dopad na softvérový projekt. V tejto eseji sa zaoberám najčastejšie sa vyskytujúcimi rizikami v softvérových projektoch, spôsobmi ako ich správne identifikovať, ako sa s nimi čo najefektívnejšie vysporiadať a ako tento boj s rizikami vplýva na celkový beh softvérového projektu.

Úvod

Väčšina softvérových projektov sa vyvíja v nepredvídateľnom prostredí, v ktorom sa tieto projekty stretávajú s mnohými úskaliaми, ktoré môžu ovplyvniť úspešnosť projektu. Analýzy preukázali, že množstvo z vyskytnutých problémov bolo predvídateľných.

Projekt je označovaný za úspešný pokiaľ úspešne splňa požiadavky, je dodaný včas a v rámci rozpočtu. Iba jedna šestina všetkých projektov je dokončených včas a v rámci rozpočtu, jedna tretina projektov je zrušená a vyše polovica nemohla byť dokončená v [1].

Z Boehmovej štúdie [2] vyplýva, že väčšina projektov zlyháva manažérsky, nie technicky. Manažment rizík je nástroj, ktorý sa pokúša formalizovať riziko spojené s vývojom softvéru a transformovať tieto poznatky do aplikovateľného súboru princípov a praktík. Využíva pritom rôzne techniky a postupy na identifikáciu, analýzu a ovládanie rizík. Manažment rizík je zameraný na prijímanie opatrení na zabránenie vzniku rizika, prípadne minimalizáciu jeho dopadu. Zohľadnenie rizikového manažmentu v projekte, znižuje jeho vystavenosť rizikám a tým pádom zvyšuje kvalitu softvéru a zlepšuje softvérový vývoj.

„Aj keď nikdy nevieme presne predpovedať budúcnosť, môžeme aplikovať praktiky štruktúrovaného rizikového manažmentu na nahliadnutie za horizont na pasce, ktoré na nás číhajú a vykonať úkony minimalizujúce šancu alebo dopad týchto potencionálnych problémov.“ [3].

Ciele manažmentu rizík sú:

- Identifikovať riziká späté so softvérovým projektom
- Zoradiť ich podľa dôležitosti a pravdepodobnosti výskytu
- Identifikovať aktivity, ktoré musia byť vykonané manažérmi na minimalizáciu identifikovaných rizík

Riziko

Riziko je možnosť utrpieť stratu, poškodenie, nevýhodu alebo zničenie. Hodnota rizika sa dá vyjadriť vzťahom (1)

$$V(A) = C(A) * P(A) \quad (1)$$

v ktorom $P(A)$ predstavuje pravdepodobnosť, že udalosť nastane a $C(A)$ znamená cenu nákladov [4]. Hodnota rizika predstavuje očakávané straty, ktoré nastanú ak sa riziko prejaví.

Tento vzťah je jednoduchý a intuitívny, ale dá sa uplatniť len na jednoduché druhy rizík. Hodnoty ostatných rizík, ktoré sú na sebe vzájomne závislé a ovplyvňujú sa, nie je tak jednoduché vyjadriť.

Riziká softvérového projektu

Riziká striehnu vždy a všade a softvérové projekty nie sú žiadnou výnimkou a keďže je toto odvetvie relatívne nové a čoraz viac sa komplikujúce, zahrňuje rizík viac než dosť. Keil a spol. na základe štúdie [3] identifikovali nasledujúce riziká (usporiadané podľa dôležitosti):

- Nedostatok zainteresovanosti vrcholového manažmentu do projektu
- Nezainteresovanosť zákazníka
- Neporozumenie požiadavkám
- Nedostatok angažovanosti používateľa
- Neúspech splniť požiadavky používateľa
- Zmena rozsahu/cieľov projektu
- Nedostatok požadovaných znalostí/zručností/skúseností vývojového tímu

- Nedostatok zmrazených požiadaviek
- Vovedenie novej technológie
- Nedostatočné/nevhodné obsadenie pozícií v tíme
- Nezhody medzi oddeleniami v organizácii zákazníka

Kategorizácia podľa dôležitosti a možnosti ovplyvnenia výskytu rizikovej udalosti

Jedným z najzaujímavejších zistení štúdie Keila a spol.[3] bola skutočnosť, že najdôležitejšie riziká sú mimo moci manažéra. Výsledkom uvedenej štúdie je aj rozdelenie rizík do štyroch kvadrantov podľa ich dôležitosti a možnosti ovplyvnenia.

relatívna dôležitosť rizika	vysoká	1 Používateľ	2 Rozsah a požiadavky
	stredná	4 Prostredie	3 Vykonanie
		nízka	vysoká
		možnosť vplyvu a riadenia	

Tabuľka 1. Kategorizácia podľa dôležitosti a možnosti ovplyvnenia výskytu rizikovej udalosti

Kvadrant 1 – Používateľ

Do tohto kvadrantu spadajú najdôležitejšie riziká. Riziká z tohto kvadrantu vyžadujú stratégie orientované na vytvorenie a udržanie dobrých vzťahov so zákazníkom a zvýšenie zainteresovania zákazníka. To si vyžaduje získanie dôvery zákazníka a to sa dá dosiahnuť pravidelnými stretnutiami. Ale získanie počiatočnej zaangažovanosti nestačí, aby nenastala situácia, že zrazu sa záujem a podpora zákazníka vytratí, je nutné neustále zvyšovať a udržiavať zákazníkovu zaangažovanosť.

Ďalším významným problémom v tomto kvadrante sú prehnané očakávania zákazníka, a preto je nutné zamerať sa na koordináciu týchto nerealistických očakávaní.

V skratke, riziká z tohto kvadrantu nemôžu byť manažérom ovládané, ale môže ich ovplyvniť. Manažovanie rizík z kvadrantu 1 zahŕňa manažment vzťahov, budovanie dôvery a politické schopnosti.

Kvadrant 2 – Rozsah a požiadavky

Aj keď je veľká snaha už od počiatku presne definovať rozsah a presné požiadavky projektu, je to vo väčšine prípadov nemožné a presná špecifikácia sa formuje postupne a používatelia dokážu presnejšie definovať čo ma daný systém robiť.

Jedným zo zaujímavých prístupov je definovať, čo systém robiť nebude. V snahe vyhnúť sa veľkým zmenám rozsahu projektu by mali manažéri informovať zákazníkov o tom, aké dopady na rozpočet ako aj časový plán takéto zmeny majú. Na ochranu pred rizikami kvadrantu 2 musia manažéri definovať požadovanú a úplne nevyhnutnú funkcionálnu vyvíjaného softvéru a stanoviť hranicu medzi nimi.

Kvadrant 3 – Vykonalie

Príkladom na rizikové faktory z tohto kvadrantu sú úskalnia spojené so slabým projektovým manažmentom, prijímaním nevhodných zamestnancov, nedostatkom metodológie, slabým odhadom a nesprávnym rozdelením úloh a zodpovednosti v tíme.

Riziká, ktoré spadajú do tohto kvadrantu silne súvisia so skúsenosťami manažéra a skúsení manažéri prikladajú týmto rizikám nižšiu dôležitosť ako manažéri, ktorí s týmito rizikami nemajú ešte také skúsenosti [1].

Kvadrant 4 – Prostredie

Riziká tohto kvadrantu sa týkajú projektového prostredia, ktoré sa nachádza vnútri aj mimo organizácie.

Tieto riziká zahŕňajú zmenu rozsahu/cieľov projektu (zmenou v manažmente alebo podniku celkovo) a problémy, ktoré môžu vzniknúť medzi oddeleniami. Riziká v tomto kvadrante sa vyznačujú tým, že na ne nemá manažér prakticky žiaden vplyv a šanca na ich výskyt je relatívne nízka, avšak keď sa tak stane sú výrazne nebezpečné.

Spôsoby prevencie a vysporiadania sa s rizikami

Manažment softvérového projektu začína nadobúdať v mnohých organizáciách čoraz podstatnejšie postavenie. S novými rizikami vyskytujúcimi sa pri implementácii projektov, musia byť použité formálne metódy na ich regulovanie.

Ak nie je vo vývoji použitý žiaden formálny nástroj na komunikáciu, ľudia inklinujú k vykazovaniu priaznivých informácií a zatajovaniu tých nepriaznivých [1], čo môže danú situáciu len zhoršiť a dáva možnosť vzniku ďalších rizík a v krajnom prípade to lavínovým efektom spôsobí zlyhanie celého projektu.

Vývoj a dodržiavanie plánu

Podľa vedeckých odporúčaní by mal byť do všetkých plánov softvérových projektov zahrnutý zoznam všetkých známych a potenciálnych rizík, pričom je vhodné použiť aj mechanizmus prioritizácie pre lepšie zameranie pozornosti.

Mne osobne sa javí táto stratégia vysporiadavania sa s rizikami ako najvhodnejšia, keďže zahrnutím rizík priamo do plánov sa tieto riziká stávajú viditeľnejšími a nakoľko platí, že na to, na čo priamo vidíme, si dávame výrazne väčší pozor než na riziká, ktoré si neuvedomíme, prípadne opomenieme, mal by tento prístup výrazne prospieť vývoju softvérového projektu.

Kombinovanie vnútorných vyhodnotení s externými revíziami

Kvôli udržaniu projektu v správnych koľajách by sa mali vnútorné vyhodnotenia kombinovať s externými revíziami, keďže existuje možnosť neobjektívneho vnútorného hodnotenia. Kľúčom k úspechu je rozoznať rizikové oblasti dostatočne skoro na to, aby mohli manažéri vykonať potrebné protipatrenia.

Zainteresovanie manažmentu počas celého životného cyklu projektu

Projektoví manažéri by mali byť schopní vytvárať a udržiavať vzťahy s používateľmi a zvýšiť tak angažovanosť používateľov v projekte.

Zainteresovanie používateľov počas celého životného cyklu projektu

Manažéri by nemali rozširovať projekt bez konzultácie s používateľmi a bez revízie rozpočtu a plánov, lebo hrozí riziko, že vývojári sa sústredia na funkcionality, ktorá nie je pre používateľa tak podstatná na úkor niečoho, čo on považuje za významnejšie.

Zabezpečiť prítomnosť riadiacej komisie

Vo väčšom softvérovom projekte by sa mala zriadiť funkcia riadiacej komisie, ktorá by dohliadala na vývoj a dodržiavanie plánov. Pre zvýšenie efektivity by táto komisia mala ultimátnu autoritu. Mali by sa konať pravidelné schôdze slúžiace na preukázanie postupu a taktiež na oznámenie zmien, dodatkov a prípadných náprav.

Rozdelenie zodpovednosti medzi členov tímu

Človek, ktorý za niečo nesie zodpovednosť sa zväčša správa výrazne odlišne ako ten, čo zodpovednosť nenesie. Pri takomto rozdelení zodpovednosti sa vlastne členovia tímu kontrolujú vzájomne a neustále a môžu tým pádom reagovať okamžite, čo môže výrazne urýchliť vývoj.

Podľa môjho názoru je tento bod absolútna nutnosť v každom tíme, nakoľko zamestnanec, ktorý nenesie žiadnu zodpovednosť nemá žiadne praktické dôvody na to aby na blížiaci, prípadne vzniknuté riziko upozornil, alebo sa pokúsil vykonať nejaké opatrenia a môže sa stať, že len tak sleduje vznikajúcu katastrofu a dbá len na to aby on vykazoval výkon.

Vyvinúť plány pre správu personálnych problémov

Presný návrh potrebného počtu a zloženia zamestnancov sa ukazuje ako absolútne kritický vo vývoji softvéru. Veľký počet zamestnancov môže viesť k väčšej komunikácii a náročnejšej koordinácii, čo sa prejaví na nižšej produktivite. Na druhej

strane však nízky počet zamestnancov často vedie k omeškaniam, nestálym prioritám a neadekvátnemu testovaniu. Obvyklým javom je, že počas životného cyklu projektu sa počet zamestnancov zvyšuje. Podstatným faktorom je nielen počet, ale aj zloženie ľudí v tíme z hľadiska ich povahových charakteristík.

Zahrnúť formálny a periodický odhad rizík

Odhad rizík pozostáva z troch aktivít: identifikácia rizík, ich analýza a priradenie priorit. Pomocou analýzy je jednoduchšie pochopiť kedy, prečo a ako sa môžu riziká prejaviť.

Pre väčšinu softvérových projektov je odhad rizík založený na neformálnych procesoch. Veľmi málo organizácií vykonáva pravidelné a formálne odhady rizík, aj napriek tomu, že prakticky všetky spoločnosti využívajú určitú metodológiu vývoja softvéru. Problém je v tom, že odhad rizík je chápaný ako ďalší proces a nie ako pevná súčasť metodológie.

Rozdeliť projekt na ovládateľné časti

Rozdelenie projektu na menšie časti je známy postup pri riadení väčších projektov a taktiež sa odporúča z hľadiska riadenia rizík, nakoľko je jednoduchšie spravovať menší počet rizík v malej časti ako sledovať riziká v celom projekte a aj v prípade keď riziko vypukne v rámci jednej časti, je menšia šanca ovplyvnenia ostatných častí.

Stanoviť rozmer dopadu zmien požiadaviek a špecifikácie na rozpočet a časový plán

Na vyhnutie sa problémom so zmenou rozsahu projektu by mali manažéri informovať používateľov o dopade týchto zmien na rozpočet a časový plán. Projektoví manažéri by mali byť schopní rozlišovať medzi požadovanou a nevyhnutnou funkcionalitou.

Stabilizovať požiadavky a špecifikáciu tak skoro ako je to len možné

Mnoho projektov sa stretáva s neistotou, keď sú prvýkrát stanovené požiadavky. Projektový manažér by mal prejsť svoje schopnosti a zabezpečiť, aby používatelia pevne stanovili detailné požiadavky tak skoro ako je to len možné.

Vyhnuť sa prehnanému počtu nových funkcií v softvérovom projekte

Je výhodnejšie novú funkcionalitu softvéru pridávať postupne, ako sa snažiť o pridanie veľkého počtu nových funkcií súčasne [1].

Posúdiť pokrok k danému termínu a stanoviť ciele na ďalšie obdobie

Skutočný stav projektu by mal byť často porovnávaný s projektovým plánom, je tak možné rýchlo objaviť problém a taktiež sú, podľa môjho názoru, krátkodobé ciele pre zamestnancov viac motivujúce ako dlhodobé. Príkladom môže byť situácia, keď zamestnanec danú úlohu v danom období nevykoná. V prípade pravidelných stretnutí je okamžite súrený danú úlohu vykonať, prípadne sú z jeho správania odvodené

dôsledky. Pri dlhodobých cieľoch sa tento deficit prejaví výrazne neskôr (čo môže byť aj dôvodom, prečo to daný človek nevykonal). Pri takto neskorom zistení môže mať takýto problém výrazne vyšší dopad na celý projekt.

Záver

Manažment rizík by mal byť jednou z hlavných častí projektového manažmentu, ale napriek tomu sa mu nevenuje dostatočná pozornosť a s rizikami sa kalkuluje len tak prirodzene, bez použitia formálnych nástrojov a postupov na ich presnú identifikáciu, analýzu a riešenie. A toto sa deje aj napriek faktu, že jediné riziko môže spôsobiť úplný zánik projektu.

Manažment rizík by sa mal stať pevnou súčasťou vývoja softvéru v každej jeho životnej fáze, byť chápaný ako jedna z podstatných súčastí vývoja a nie ako práca navyše. Manažéri by si mali uvedomiť, že v súčasnej dobe dravej konkurencie, kde každý vyčkáva na zakopnutie toho druhého, sa neoplatí hrať ruskú ruletu a len dúfať, že v komore náboj nebude a jemu sa podarí vyhnúť nástrahám, ale aktívne sa snažiť minimalizovať riziko a tým maximalizovať šance na dotiahnutie projektu k úspešnému koncu.

Použitá literatura

1. Addison, T., Vallabh, S.: Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers. *Proceedings of SAICSIT (2002)*, 128 – 140.
2. Boehm, B.W.: Software risk management: principles and practices. *IEEE Software*, Vol. 8, No. 1 (1991) 32-41.
3. Keil, M., Cule, P.E., Lyytinen, K. and Schmidt, R.: A Framework for Identifying Software Project Risks. *Communications of the ACM*, Vol 41[11], 77-83.
4. Roy Geoffrey G.: A Risk Management Framework for Software Engineering Practice, In: *aswec*, p. 60, 2004 *Australian Software Engineering Conference (ASWEC'04)*

Annotation

Software project risk management

Risk, most specifically such, that can abort a work-in-progress project is something that has to be taken care of seriously. Even so, software project failure is surprisingly common. In most cases, failure could be prevented by adhering to certain principles and methods collectively called risk management. Risk management covers procedures how identify, analyze and eliminate risks or minimize their occurrence and effect on software project. In this essay I discuss the most frequently occurring risks in software projects, ways to their identification, most effective methods of their elimination and overall impact of this struggle on software project development.