

AKO NA ROZMAZNANÝCH ZÁKAZNÍKOV

Najviac nahnevá neočakávaný dopad rizika, ktorého vznik sa dal očakávať.

Adam Mihalik

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
zegee29@gmail.com

Abstrakt. *V dnešnej dobe pociťujeme rózny vývoj informačných technológií, no stále nie sme schopní čeliť všetkým rizikám projektu. Potrebné je uvedomiť si nielen riziká samotné, ale snažiť sa o elimináciu najmä tých, pre ktoré má projekt najväčšiu slabosť. Esej prináša pohľad na riziká týkajúce sa zákazníka ako jedny z najnebezpečnejších rizík projektu. Pokúsime sa určiť úspešnosť často používaných metód eliminácie rizika, pričom ich zároveň vylepšíme pre riešenie konkrétnych problémov týkajúcich sa požiadaviek zákazníka. Analyzujeme úspešnosť najčastejšej metódy identifikácie rizík a porovnáme s úspešnosťou aplikovania metódy dekompozície na rovnaký problém. Pozdvihneme dekompozíciu ako metódu pre určenie rizikovosti konkrétnej oblasti projektu, vďaka ktorej je následne omnoho jednoduchšie predpovedať vznik a zmeny požiadaviek zákazníka.*

Kľúčové slová: *manažment rizika, riziko, zákazník, požiadavka, zber požiadaviek, dekompozícia, prototypovanie*

Úvod

V dnešnej dobe jasne pociťujeme, ako dynamicky a rózne sa vyvíja prostredie informačných technológií. Avšak ani pomocou moderných nástrojov pre riadenie a plánovanie projektu nie sme schopní eliminovať všetky riziká s ním spojené.

Ako príklad si zoberme výsledky štúdie o úspešnosti projektov vo svete [4]. Podľa výsledkov tejto štúdie len 29 % projektov sa ukončilo úspešne, čiže neprekročili stanovený rozpočet, boli dodané relatívne načas a výsledky týchto projektov boli na prijateľnej úrovni

z pohľadu zákazníka. So zvyšnými projektmi to už nebolo také jednoduché. Až 18 % projektov skončilo fiaskom. Buď bol neakceptovateľne prekročený čas ich dodania alebo výrazne prekročili disponibilné náklady.

Je táto štatistika uspokojivá? Množstvo neúspešných projektov sa nezdá ničím horibilným, ak v úvahu berieme neustále meniace sa technológie a požiadavky zákazníkov. Podľa môjho názoru je ale úplne neprijateľné, ak ani výsledky aj takýchto projektov nie sú použiteľné v iných, podobných projektoch. Môj názor nemením ani na zvyšnú časť - 53 % zo všetkých projektov. Ich výsledky síce boli dodané v prijateľnom čase, ale tento čas sa od pôvodného výrazne líšil a nebolo to inak ani so zbytočným prekročením predpokladaného rozpočtu. V podstate, štyri pätiny všetkých projektov boli nakoniec ukončené a dodané zákazníkovi. Spreádzali ich ale privysoké náklady za realizáciu. Pre veľké firmy nebýva často problém akceptovať zvýšené náklady alebo nedosiahnutie požadovaného zisku, ale pre malé firmy to môže mať fatálne následky. Avšak nepovažujem tieto rizikové situácie za neriešiteľné, aj keď komplexnosť projektu evokuje vznik rôznych rizík. Čo je ale príčinou vzniku týchto problémov? Ak ich aspoň včas odhalíme, dokážeme sa vyvarovať chybám, ktoré by ich mohli spôsobiť?

Cieľom eseje je teda preskúmať príčiny vzniku častých problémov pri vývoji projektu. Bližšie sa budeme zaoberať najmä rizikami spojenými so zákazníkom a jeho požiadavkami a či je vôbec možné aspoň čiastočne eliminovať ich dopad.

Za horami, za dolami, stalo sa raz jednej firme

„Kde sme spravili chybu? Veď sme dobrí v tom, čo robíme...“, povedal si manažér projektu istej firmy XYZ v jedno ráno, keď si uvedomil neúspechy projektu. Aké že ale neúspechy? Veď ako naznačil, sú dobrí v tom, čo robia. Firma XYZ zamestnáva kvalitných špecialistov, skúsených z podobných projektov. Dokonca títo zamestnanci pracujú v rokmi overenom hierarchickom rozčlenení menšieho tímu. Morálka nikdy neklesla, nikto z tímu neodišiel, ani nezomrel. Definovaniu možných rizík sa zaoberali tak, že spisovali všetky riziká, s akými sa doteraz stretli. Tak kde mohla nastať chyba?

Odpoveď je jednoduchá. Zlyhal manažment rizika. Pri hľadaní častej otázky „Kde sa stala chyba?“ vystihnutú v opísanej situácii firmy XYZ môžeme akýsi základ nájsť v štúdiu [2], kde autori rozdelili zoznam rizík do štyroch kvadrantov rozložených podľa úrovne ich dôležitosti a kontroly. Nateraz by som chcel vyzdvihnúť a dať do povedomia *kvadrant týkajúci sa hlavne zákazníka*. Uvádza sa, že *riziká spojené so zákazníkom majú najvyššiu prioritu riešenia a zároveň sú najmenej ovplyvniteľné rozhodnutiami vedenia projektu*. Riziká umiestnené v tomto kvadrante zahŕňajú, okrem problémov implementácie nesprávneho rozhrania, aj problémy s manažmentom zberu požiadaviek. V oblasti celého manažmentu rizika sa tak jedná, čo mi logika našepkáva, o najnebezpečnejšiu, najzložitejšiu a nepochybne najnepredvídateľnejšiu časť projektu. A popritom je to všetko také dôležité.

Firma XYZ síce má pozitívne skúsenosti spisovaním zoznamu možných rizík, ale predpokladám, že priveľmi podcenili manažment rizika práve v oblasti zákazníckeho mandátu. Ich spôsob identifikovania rizík napríklad neuvažuje, že zákaznícka firma v skutočnosti nevie, čo chce. Vlastne ani jedna firma nevidela do procesov a nechápala logiku tej druhej firmy. A tak sa jednoducho stalo, že sa pri zadávaní požiadaviek často krát nepochopili, alebo množina požiadaviek sa časom neúmerne zväčšovala. Niekedy bol

proces omnoho zložitejší, ako to podal zákazník, priam až takmer nedosiahnuteľný. Inokedy si zákazník rozmyslel svoje rozhodnutie a prezentoval inú požiadavku alebo apeloval na zmenu existujúcej. Dalo sa ale tomuto predísť? Môj názor je, že áno.

Čo je rizikom dnes

Viacero odborníkov z oblasti manažmentu rizík podkladá Boehmov zoznam stále za aktuálny [1]. Boehmova kategorizácia rizík je pomerne dobrá, ale niektoré kategórie by sa dali zlúčiť pre ich lepšie spracovanie manažmentom rizík. Pre našu problematiku som sa zameral hlavne na zlúčenie rizík týkajúcich sa priamo zákazníka, pričom som vychádzal zo štúdie [3] kde zlúčili riziká do oblastných komponentov. Táto štúdia sa síce zaoberala rizikami veľkých projektov, ale niektoré zistenia mi pripadajú dobre aplikovateľné aj na menšie projekty. Riziká súvisiace so zákazníkom boli zlúčené ako komponenty:

- Systémová funkcionálnosť - vznikla zlúčením „Vývoja zlej funkcionality“ a „Vývoja zlého používateľského rozhrania“.
- Manažment požiadaviek - je zlúčením „Pozlúčenia systému“ a „Neustálej zmeny požiadaviek“.

Ako na problém zle implementovaného rozhrania

Vystihnúť problematiku nesprávnej implementácie funkcionality sa dá pomocou vtipnej situácie:

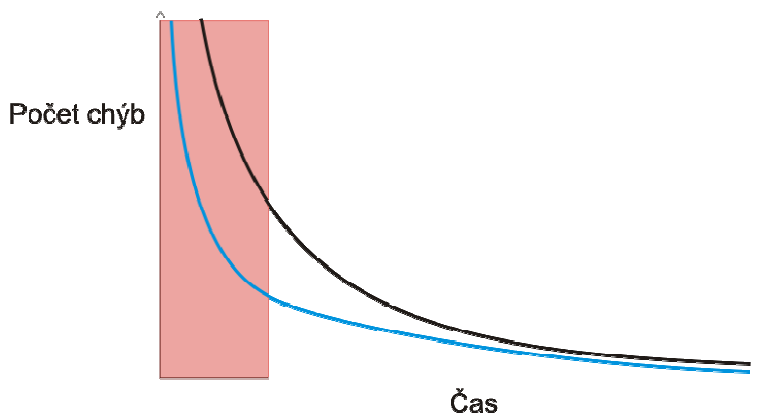
„Zákazník má ideu. Svoju ideu prezentuje programátorovi. Programátor si predstaví, ako ideu spracovať do reálnej podoby. Programátor uistí zákazníka, že vie presne na čo myslí. Programátor implementuje ideu do aplikácie. Aplikácia sa prezentuje zákazníkovi. Zákazník ostane vyjavený, čo to je?! Spíše posudok spokojnosti s aplikáciou. Programátor si posudok prečíta a ostane vyjavený, čo to je?!“

Darmo, iný kraj, iný mrav, iný jazyk. Ani častá komunikácia so zákazníkom nemusí odhaliť podstatu jeho požiadavky. Nesprávne pochopenie požiadavky môže spočívať práve v spôsobe komunikácie, kde sa zákazník a vývojár vyjadrujú „iným jazykom“. *Najlepším komunikačným jazykom zákazníka a vývojára je skoré prototypovanie a jeho skúšanie.* Tým by sa malo značne znížiť riziko nesprávnej implementácie alebo rozhrania aplikácie.

Evolučný proces tvorený cyklami vytvorenia prototypu a jeho následného hodnotenia zákazníkom mi ale pripadá mierne neefektívne. Mojim argumentom je, že hlavne na začiatku vývoja môže byť pripomienok na strane zákazníka pomerne dosť. Po prezentovaní ďalšieho prototypu, ktorý vznikol opravou predošlého, je už pravdepodobne menej pripomienok na zmenu. *Navrhujem preto „preskočiť“ úvodné prototypovanie plné nesprávnych funkcionálností a nahradiť ho za menej nákladný spôsob návrhu rozhraní.* Vplyv opísaného nahradenia na celkové náklady je výstižne opísaný grafom (pozri Obr. 1). Čierna krivka reprezentuje počet chýb vzniknutých pri prezentovaní prototypom, modrá krivka reprezentuje počet chýb vzniknutých pri dočasnom nahradení prototypu za menej nákladný spôsob návrhu rozhraní. Červená oblasť na začiatku znázorňuje dobu, kedy sa použije náhrada implementácie prototypu. Z grafu vyčítať, že implementácia prototypu v každom cykle vyžaduje čas na realizáciu. Prezentovanie tak začína neskôr a zároveň

4 Adam Mihalik

oprava chýb postupuje pomalšie ako pri náhrade. Pri konci doby nahradenia je síce potrebné implementovať funkčný prototyp, ale predošlá dynamika práce nám prináša osoh v podobe ušetrného času.



Obr. 1. Grafová reprezentácia nahradenia vývoja prototypu.

Výhodným spôsobom dočasného nahradenia by mohlo byť napríklad vytváranie skíc, ktoré tvorí sám zákazník. Dokonca, v určitých oblastiach je nám už dnes dostupná technológia priameho vytvárania plne funkčného rozhrania aplikácie. Predpokladám, že použitím takýchto nástrojov by si zákazník uvedomil prípadné nezmyselnosti svojich návrhov skôr a programátor by zároveň dostal presnú špecifikáciu výstupu.

Uvedomme si význam tohto „zefektívnenia“ návrhu. Ako vývojári nebudeme nútení prototypovať aplikáciu stále dookola, ale až v takmer finálnej fáze, kedy si budeme dostatočne istí návrhom riešenia či už funkcionality, rozhrania alebo celkovej architektúry. Ale prijal by zákazník takýto štýl zberu požiadaviek?

Ako na problém s požiadavkami zákazníka

Aj napriek nie veľmi bohatým skúsenostiam s vedením projektu, som si všimol jednu podstatnú vec. Celý projekt a jeho výstup sa týka hlavne okolo požiadaviek a potrieb zákazníka. Neexistuje správny alebo nesprávny zákazník. Neexistuje ani zlý alebo dobrý zákazník. Zákazník sa charakterizuje hlavne typom jeho požiadaviek a čo všetko naozaj chce. Niekedy je zákazník ako malé dieťa, ktoré nevie čo chce a predsa chce všetko. Niekedy je ako džentlmen, ktorý vie presne čo chce, ale nevie, ako sa to dá dosiahnuť a odovzdá sa do rúk nám, vývojárom. Pri takto definovaných požiadavkách nastáva riziková situácia, pretože požiadavky sú často nejasné, nepresné. Absolútne tak nie je zaručená stálosť množiny požiadaviek.

Úlohou manažmentu rizík je špecifikovať spolu so zákazníkom nejasné požiadavky a zároveň zabrániť zásadným zmenám v existujúcich požiadavkách [3]. Stretnúť sa s častou zmenou požiadaviek nie je vôbec problém. Napríklad zákazník požaduje zobrazovanie detailu zloženia pracovnej skupiny. Potom požaduje zaznamenávanie zmien jej zloženia v čase. Potom usúdi, že by nebolo zlé, keby mohol automaticky „presúvať“ zamestnancov medzi skupinami. Dokedy bude zákazník so zmenou požiadaviek pokračovať? Skončí niekedy?

Ak ho nezastavíme, môže nám to poriadne sťažiť vývoj. Myslím si preto, že nie je správne mu dávať príliš po vôli. Reálne sa stal prípad, že nakoniec si zákazník uvedomil nezmyselnosť evidencie pracovných skupín a túto funkcionálnu v podstate zavrhol a zahodil tak 2 mesačnú prácu. Zákazník sa mnohokrát „vyspí“ zo svojich predstáv a potom usúdi, že by sa to dalo navrhnúť aj inak.

Ale ako eliminovať vznik tohto rizika? Ako sme už rozoberali, zákazník je v podstate niekto, koho zaujíma najmä výstup projektu. A keď ho už vidí, nechce vidieť nič iné, ako to, čo očakával, že uvidí. Inak je nespokojný, pretože vo svojom názore je neovplyvniteľný. Nie je preto možné nájsť žiadnu formálnu cestu, ako eliminovať zdroje rizík pri zbere požiadaviek. Preto si dovoľím doplniť, pre dobrý manažment zberu požiadaviek, zodpovednosť aj za predpovedanie vzniku nových požiadaviek a ich zmien.

Po bitke je každý generál, ale dalo by sa to zachrániť?

Vrátim sa späť k prípadu firmy XYZ, pričom použijem znalosti, ktoré sme nadobudli. Riziká zákaznickeho mandátu môžu vznikáť a meniť sa neustále, najmä ak zákazník nemá ucelenú predstavu o výsledku. Preto metóda vytvorenia zoznamu rizík nie je použiteľná v úplnej miere, akú vo firme uvažovali. Na „statické“ riziká v oblasti vnútorných procesov firmy, ako je napríklad použitá technológia alebo dostupnosť ľudských zdrojov, sa spíše zoznam rizík na základe charakteru projektu a skúseností vedenia. Ale čo sa týka všetkých problémov s požiadavkami zákazníka, išlo by o hľadanie ihly v kope sena. Pripomínam, že v ich prípade bol najväčším kameňom úrazu to, že zákazník a firma nepoznali svoje procesy navzájom. Vyzdvihol by som preto použitie metódy dekompozície [1] ako úspešnejšie v procese identifikácie rizík spojených s dynamicky sa meniacimi požiadavkami zákazníka.

Samotná metóda dekompozície len napomáha k lepšej identifikácii rizík tým, že určí najmenšie funkcionality komplexného systému. Považujem ale za výhodné, použiť jej výstupy skombinované s návrhmi identifikácie rizík, ktoré sme si uviedli ako príklady riešenia v predošlých častiach eseje.

Výhody dekompozície pre projekt a výhody pre manažment rizika pre zjednodušenie uvádzam v tabuľke 1.

Tab. 1. Výhody dekompozície pre vývoj projektu a z toho vyplývajúce výhody manažmentu rizika.

Výhoda pre vývoj	Výhoda pre manažment rizika
vedenie projektu má príležitosť úplne pochopiť všetky procesy, ktoré zákazník potrebuje aplikáciou ovládať	zamedzilo by to akýkoľvek pokus o vytvorenie nesprávnej funkcionality. Eliminácia výskytu rizika pozlátenia systému
aplikovanie prototypovania na menšie, procesne jednoduchšie časti	možnosť zefektívnenia navrhovania rozhraní pomocou nástrojov aspoň pri jednoduchších častiach
možnosť vzniku návrhu architektúry riešenia s podrobným rozdelením podľa funkcionálnych elementov	lepšie predpovedanie zmeny požiadaviek pre konkrétnu funkcionálnu podoblasť aplikácie

Vďaka dekompozícii môžeme čerpať z jej výhod a to aplikovaním už spomenutých návrhov na elimináciu vzniku rizík zo strany zákazníka. Je nám ale poskytnutá nová možnosť a to zľahčenie predpovedania zmien v existujúcich požiadavkách a možno aj predpovedania vzniku nových. *Ak máme návrh architektúry riešenia produktu s dostatočne nízkou granularitou, dokážeme sledovať frekvenciu zmien v požiadavkách týkajúcich sa konkrétnej podoblasti a určiť tak kritickú oblasť.*

Predstavme si príklad, že zákazník stále mení požiadavky týkajúce sa vytvárania výkonov pracovnej skupiny, ale požiadavka o editácii samotnej pracovnej skupiny sa nezmenila od prvotného návrhu. Z toho vieme usúdiť, že s vyššou pravdepodobnosťou sa bude ďalšia zmena týkať funkcionality denných výkonov. Následne, ako manažment rizika, vieme upozorniť vedenie o nestabilite danej funkcionality a každý iný manažment sa presnejšie zameria na podoblasť po svojom. V kritickvej oblasti tak vývojári radšej nebudú riskovať zbytočným implementovaním nadbytočnej funkcionality, analytici sa budú podrobnejšie pýtať na vznik požiadaviek a návrhári sa v tejto oblasti už radšej spýtajú, ako riešiť vzniknutú nejasnosť a nebudú prezentovať svoje vlastné návrhy.

Záver

Rizík sa netreba báť. Rizík sa treba obávať. Neexistuje projekt, ktorý je bez menších-väčších rizík. Je ale podstatné, práve kvôli tomuto všetkému, nezatracať manažment rizika ako niečo nepotrebné. Málokto ma rád nadpriemerné opatrenia, kvôli ktorým by sme sa nenajedli ani z vlastného taniera. Ale nie je ani správne držať sa programátorskej zásady „mne to ide a to stačí“. Práve udržanie sa na zlatej strednej ceste je správnym spôsobom ochraňovania vývoja projektu až do konca. A kedy spoznáme, že robíme správne? Kedy spoznáme, že sme sa stali odborníkmi manažmentu rizika? Na otázku môžeme prestať hľadať odpoveď, ak zabránime rizikám stať sa skutočnými problémami, i keď sa jedná o tak nekontrolovateľný element projektu, akým je aj sám zákazník.

Použitá literatúra

1. Barry W. Boehm: *Software risk management, Principles and Practices*, IEEE Software, Vol. 8, 1991.
2. Mark Keil, Paul E. Cule, Kalle Lyytinen, Roy C. Schmidt: *A Framework for identifying Risks*, Communications of the ACM, November 1998.
3. Ropponen, J., Lyytinen, K.: *Components of software development risk, How to Address them ? A project manager survey*, In: *IEEE transactions on software engineering*, 2000.
4. Stephen P. Masticola: *A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management*, Siemens Corporate Research, 2007.

Annotation

How on pampered customers

The essay points out, that despite the advanced development of information technologies we are not able to meet all project risks. At the same time it provides insight into the risks to the customer as one of the most dangerous risks of the project. We determine the percentage of most frequently used methods of the risk elimination and we upgrade them for a finding the specific issues related to customer requirements. We analyze the success of the most common methods of identifying risks and compared with the success of applying decomposition method to the same problem. Using the decomposition method we determine the riskiness of specific project areas, thus we can predict the formation and changes in customer requirements.