

MANAŽMENT RIZÍK V ŠKOLSKOM SOFTVÉROVOM PROJEKTE

Najväčšie riziko je neznáme riziko.

Michal Macko

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

Abstrakt. *Takmer každá činnosť je spojená s určitým rizikom. Výnimkou samozrejme nie je ani softvérový alebo akýkoľvek iný projekt. Toto riziko môže byť rôzne veľké a mať rôzne následky, od nepatrných prekážok až po zlyhanie celého vynaloženého úsilia. Na druhej strane riziko znamená výzvu, ktorá môže, ale nemusí prinášať zisk. Preto je potrebné vedieť o rizikách, ktoré môžu počas práce na projekte nastať a snažiť sa ich minimalizovať, prípadne nájsť ideálnu hranicu medzi ohrozením projektu a získaním nejakej výhody. V tejto eseji spomeniem časté rizikové faktory a zameriam sa na tie najdôležitejšie z nich z pohľadu nášho školského tímového projektu. Taktiež uvediem spôsoby znižovania a odstraňovania týchto rizík.*

Kľúčové slová: *project risks, risk analysis, risk management, risk planning, risk factors*

Úvod

Projekty bývajú rozsiahle, komplexné, majú veľké množstvo požiadaviek a vyžadujú si značné úsilie na ich úspešné dokončenie. Preto nie je isté, či sa to vôbec podarí. Niektoré zdroje [2, 3] uvádzajú len približne 30% úspešnosť so splnením všetkých požiadaviek, iné 20 – 50% [1]. Podľa [3] až 53% projektov čelí problémom, ktoré môžu spôsobiť presiahnutie rozpočtu, nedodržanie plánu, alebo menšiu funkcionality výsledného riešenia [2]. Zhruba 20% projektov je úplne zrušených ešte pred dokončením [3].

Myslím, že na každom projekte pracuje tím zložený zo schopných odborníkov. Ako je teda možné, že toľko z nich skončí bezúspešne, alebo len s čiastkovým úspechom? Veď predsa určenie rozpočtu, požiadaviek, časového plánu a iných vecí je základ, ktorý nikto nepodceňuje a snaží sa ho urobiť čo najpresnejšie a najreálnejšie. Asi to bude tým, že pri takýchto rozsiahlych projektoch je mnoho faktorov, ktoré nie sú až také zrejmé na prvý pohľad, ale majú kľúčový význam pre výsledok projektu. Práve o týchto faktoroch bude v tejto eseji reč a tiež o tom, ako s nimi pracovať, aby sme dosiahli želané výsledky.

Čo je vlastne rizikom?

Rizikom môže byť podľa môjho názoru v podstate čokoľvek. Našťastie niektoré veci sú natoľko samozrejmé, že sa už ani neberú ako nejaké riziko. Je predsa úplne jasné, že potrebujeme mať v tíme kvalitných schopných pracovníkov, nejakú technológiu a podobne. Avšak často sú rizikové rôzne „maličkosti“, ktoré také zrejmé nebývajú, alebo dokonca to môže byť prehnaná snaha o čo najlepší výsledok, ktorá môže viesť skôr k opaku. Najväčším rizikom je podľa mňa riziko, o ktorom nevieme. Na to totiž nedokážeme reagovať, predvídať ho, a teda môže napáchať najväčšie škody. Z toho logicky vyplýva potreba človeka, manažéra rizík, ktorý bude dohliadať práve na to, aby sa nič nepodcenilo v tejto oblasti. Každý člen tímu by mal mať určité povedomie o možných rizikách, ktoré dokáže ovplyvniť, ako aj o tom, ako ich znížiť alebo ešte lepšie úplne odstrániť. Ale funkcia jedného človeka, ktorý to má celé „pod palcom“, je, zvlášť pri veľkých projektoch a veľkých tímoch, veľmi cenná. To nakoniec dokazuje aj skutočnosť, že sa táto problematika prejednáva na odbornej úrovni a učíme sa o nej aj na našom tímovom projekte.

Rizikové faktory

Čo sa týka samotného identifikovania rizík a ich začlenenia do nejakých skupín alebo kategórií, na to je viacero uhlov pohľadu. V [3] sú uvedené štyri kategórie – *technické faktory*, *obmedzenie zdrojov*, *organizačné faktory* a *iné faktory*. Naproti tomu v [1] sú vypísané podrobnejšie riziká, ktoré sa mne osobne aj najviac páčia, spolu ich je až 14 a myslím si, že pokrývajú podstatnú časť procesu vývoja systémov. Článok [2] má základné členenie rizík v prehľadnej tabuľke respektíve grafe, kde vidno veľkosť rizika aj mieru jeho kontroly. Myslím, že najrozumnejšie by bolo začať najskôr s jednoduchším, širším členením a postupne prejsť k tomu podrobnejšiemu, pričom sa pokúsim zamerať hlavne na tie faktory, ktoré môžu byť podstatné pri riešení školského projektu.

Ten je trochu špecifický v tom, že na ňom pracuje malá skupinka ľudí, a teda sa nemôžu oddeliť riadiace a výkonné funkcie. Každý člen sa stará aj o manažment aj o vývoj systému, čo môže byť problematické. Rozsah projektu rovnako nemôže byť priveľký, tým pádom by sa nemali vyskytnúť problémy súvisiace s riadením obrovských komplexných projektov. Nedostatok skúseností zas na druhej strane do určitej miery koriguje projektový vedúci, ktorého úlohou je tím naviesť na správnu cestu a poradiť pri rôznych nejasnostiach.

Podme sa teda pozrieť, ako sú delené riziká v [2] (Obr. 1). Máme štyri kvadranty, ktoré sú umiestnené podľa miery rizika a úrovne kontroly. Kvadrant 1 značí faktory

súvisiace so *zákazníkmi a používateľmi*. Ich dôležitosť je síce vysoká, ale miera kontroly naopak nízka. V druhom kvadrante sú faktory, ktoré sú asi najkľúčovejšie, a to *rozsah a požiadavky*. Majú veľký dopad na konečné riešenie a rovnako aj vysokú mieru kontroly, čo z nich robí ideálnych kandidátov pre rizikových manažérov. Ďalšie takéto faktory sú zahrnuté v treťom kvadrante, tie sa týkajú samotnej *realizácie projektu*. Plánovanie, vývoj, testovanie atď. Veľkosť rizika je už miernejšia, takže manažéri sa tu až tak veľmi nenatrápia. No a v poslednom kvadrante sú faktory súvisiace s vnútorným a vonkajším *prostredím*. Tam sa nachádzajú veci, ktoré asi ani nie sú veľmi pravdepodobné a nemajú veľký dopad na projekt, ich výskyt sa taktiež veľmi nedá ovplyvniť.



Obr. 1. Kategórie rizík

Podľa môjho názoru je toto celkom užitočné rozdelenie, ktoré napomáha zorientovať sa v problematike, získať určitý nadhľad.

Ďalšia podobná už vyššie spomenutá kategorizácia je v článku [3], kde autori členia riziká na technické faktory, obmedzenia zdrojov, organizačné faktory a iné faktory. *Technické rizikové faktory* súvisia s hardvérom a softvérom, sú to vlastne tie, ktoré sa nachádzajú aj v treťom kvadrante predošlej schémy, týkajúce sa realizácie projektu. *Obmedzenia zdrojov* sa dajú okrem druhého kvadrantu natiahnuť čiastočne aj na ďalšie. Ide tu o trojuholník čas, ľudia, peniaze a rovnováhu medzi týmito tromi aspektmi. *Organizačné faktory* zahŕňajú manažment a jeho riadiace schopnosti, organizačnú štruktúru, zaangažovanosť používateľov a podobne. Kategória *iné faktory* je v podstate rovnaká ako kvadrant 4 v predošlej schéme, čiže ide o faktory vzťahujúce sa na prostredie, ale aj kultúrne, etické, či právnické faktory.

Školský projekt

Urobili sme si teda prehľad, čomu sa máme venovať a dostávame sa k podrobnejšiemu skúmaniu problematiky, ktorú si hneď aj prepojíme so školským tímovým projektom. Pomáhať si budem článkom [1], z ktorého sa okrem rizikových faktorov dajú vyčítať aj spôsoby ich riadenia.

Na prvom mieste by som uviedol *nepochopenie rozsahu a požiadaviek*, čo sú podľa mňa faktory, s ktorými sa tím stretne na úplnom začiatku. Toto rozhodne netreba podceňiť, pretože sa od toho odvíja celý projekt. Aj keď máme rozsah pre tímový projekt pomerne malý, až taká výhoda to v tejto fáze nie je, nakoľko nemusí byť hneď jasné, ktorú časť témy vlastne pokrýva. Na odstránenie takéhoto rizika platí komunikácia so zákazníkom a používateľmi. Tú v našom prípade stelesňuje náš projektový vedúci. Aj neskôr v priebehu realizácie je žiadané akékoľvek otázky a pochybnosti ohľadne požiadaviek na projekt konzultovať so zadávateľom, keďže jemu má byť „šitý na mieru“.

Ďalším dôležitým rizikom sú pre nás *neadekvátne znalosti a schopnosti*. To súvisí aj so zložením tímu, ktoré sme ale veľmi ovplyvniť nemohli. Napriek tomu by to nemalo mať nejaké závažné následky. Verím, že systém, akým sme boli do tímov zadeľovaní, je dostatočne kvalitný na to, aby žiadny tím nemal v tomto ohľade priveľké problémy. Navyše ako študenti by sme mali hruvo zvládnuť nové metodiky a technológie aspoň na základnej úrovni umožňujúcej nám vypracovanie nášho softvérového systému. Taktiež primeraný rozsah značne znižuje závažnosť tohto faktoru.

V reálnych projektoch v každom prípade treba čo najviac zapojiť *používateľov* vytváraného systému. Odozva od ľudí, ktorým je systém určený, je veľmi dôležitá z hľadiska určenia funkcionality, ktorú má systém ponúkať. V prípade našich školských projektov je to možno ťažšie, keďže sme dosť obmedzení na prostredie fakulty. Hoci aj tu sa dá niečo robiť. Vo viacerých projektoch sú používateľmi ľudia zo školy, či už študenti alebo vyučujúci a zamestnanci, prípadne priamo zadávateľa projektu a vedúci. Druhou možnosťou, ktorú sa aj snažíme využiť v našom tíme, je vytvorenie ankety na internetovej stránke produktu, kde sa môže ktokoľvek vyjadriť k jeho očakávaniam a požiadavkám. To treba samozrejme prepojiť aj s prezentáciou vytváraného produktu, aby sa o ňom dozvedelo čo najviac ľudí a vyjadrilo svoj názor.

V nasledujúcom kroku treba získané informácie pretaviť do *funkcionality systému*. Vyjadrenia zákazníka a používateľov môžu síce obsahovať aj návrhy konkrétnych funkcií, avšak podľa môjho názoru sú to vo väčšine prípadov skôr len také vlastnosti ako napríklad, aby to bolo pekné, prehľadné, jednoduché na ovládanie, aby to obsahovalo to a ono. Tu je potrebné, podľa mňa, správne vybalansovať mieru medzi dostatočným znížením rizika a získaním výhody vytvorením niečoho nového a efektívneho. Rozhodnúť sa, či sa vydáme cestou skôr konzervatívnejšou, overenou, so všetkými výhodami aj nevýhodami, alebo stavíme na novátorský prístup a pokúsime sa eliminovať tie známe nevýhody, čo ale prináša značné riziko vzniku nových problémov. To môžu byť nedostatky v samotnej funkcionalite alebo negatívny postoj používateľov, ktorým sa nové riešenie nemusí páčiť, lebo sú už zvyknutí na to zabehnuté.

Čo sa funkcií týka, veľmi častým problémom, aj keď nie mimoriadne závažným, je „*pozlacovanie*“ systému (angl. gold plating). Určite každý pozná pocit, že jeho dielo je príliš jednoduché a v podstate nič nedokáže, že sa bude dať použiť iba na jednu alebo dve veci. A tak si začne vymýšľať ďalšie a ďalšie funkcie, ktoré by mali rozširovať použiteľnosť. Avšak skutočnosť je taká, že väčšina používateľov, ktorých som aj ja typickým príkladom, používa v systéme len zo dve tri funkcie, ktoré mu bohato postačujú na to, čo potrebuje a ten zvyšok to len zbytočne zahľucuje a komplikuje. Iné je, keď je systém určený viacerým druhom používateľov a každý z nich požaduje rozdielnú funkcionalitu, ale tak isto platí, že treba implementovať len to najnutnejšie a v prvom rade to, čo zákazník požaduje.

Z ďalších rizík uvedených v článku [1] by som opísal ešte tri, ktoré nepredpokladám, že sa objavia v školskom projekte. V praxi sa však vyskytnúť môžu a myslím, že nie zriedka a môžu byť dosť nepríjemné.

Jedným z nich sú *nerealistické časové plány a rozpočet*. Vedenie podniku alebo zákazníka môžu mať tendenciu priveľmi tlačiť na vývojársky tím, aby čím skôr projekt dokončili a odovzdali funkčný systém. To ale pravdepodobne bude mať negatívny dopad na kvalitu produktu, ale tiež motiváciu pracovníkov a podobne. Nie je dobré určiť plán ani príliš voľný, čo zas projekt zbytočne predlžuje a podporuje lajdáctvo. My študenti máme plán určený pevne, overený časom. Aj keď by sme prijali trochu voľnejší, zvládnuť sa to dá. Rozpočet máme prakticky nulový, tým pádom sa nemusíme zaoberať s jeho správnym odhadovaním, tak aby sme si dokázali zaobstarať všetky potrebné záležitosti.

Neustále *pozmeňovanie požiadaviek* je tiež vecou, ktorá dokáže pribrzdiť vývoj produktu. Nedá sa zakaždým meniť niečo, čo je už hotové alebo aspoň rozpracované. Preto je dôležité špecifikovať všetky požiadavky na začiatku projektu, ešte pred fázou realizácie, alebo opísať, čo sa dá, nejakú kosť systému a zvyšné veci neskôr, keď sa už budú dať jasnejšie špecifikovať podľa kontextu. Tam je vtedy aj priestor na zváženie viacerých možností a všetkých pre a proti.

Tým tretím faktorom je použitie *novej technológie*. Myslí sa úplne niečo nové, čo ešte nebolo dostatočne použité v praxi inými spoločnosťami. Na jednej strane to môže byť aj motivujúce pre vývojárov, príjemná zmena, snaha niečo nové sa naučiť. Hlavne ak je predpoklad, že je to jednoduchšie na používanie. Samozrejme to však zo sebou prináša ťažkosti a časové zdržanie, pokiaľ si tím na technológiu navykne a naučí sa ju používať.

Zásady riadenia softvérového projektu

Poznanie všetkých možných rizík, ktoré môžu nastať počas projektu, ešte neznamená, že sa im s určitosťou vyhneme. Aspoň mne sa zdá neustále kontrolovanie nejakého zoznamu rizikových faktorov a rozpamätávanie sa, ktoré sme už spracovali, dosť chaotické a neefektívne. Určite je lepšie zamyslieť sa nad takým spôsobom riadenia projektu, ktorý by nám všetky tieto riziká umožnil jednoduchšie identifikovať a odstraňovať bez toho, aby sme sa na ne museli cielene zameriavať. Tým myslím to, že si nebudem musieť povedať: „Tak a teraz idem zisťovať možné riziká.“, ale naopak pracujem na vývoji niektorej časti systému, alebo dohliadam na riadenie projektu a keď sa niečo vyskytne, tak okamžite viem posúdiť, či sa tam náhodou nejaké to riziko neskrýva. Práve o tom budú nasledujúce riadky. Opíšem niekoľko takýchto zásad riadenia projektu (podľa [1]) a opäť sa zamyslím aj nad ich možným využitím v našom projekte.

Na začiatku je dôležité vytvoriť si *plán vývoja* a dodržiavať ho počas celého projektu. Ja by som odporučil aj vytvorenie prípadných alternatívnych vetiev v prípade, že nevieme s určitosťou odhadnúť, aká situácia môže v priebehu realizácie nastať. Plán B, plán C atď., ktorý by nám umožnil pohotovo reagovať na vzniknuté udalosti.

Žiaduce je aj overovanie, či sme stále na *vytýčenej ceste*. Kombinovať vnútorné zhodnotenia s vonkajšími revíziami, aby sme sa ubezpečili, že stále vytvárame to, čo máme zadané.

Zapojenie manažmentu a používateľov počas celého vývoja bude v našom prípade trochu špecifické, keďže všetci členovia tímu sú aj manažéri aj vývojári. Vrcholový

manažment má zastúpenie v projektovom vedúcom. Angažovanie používateľov bude asi závisieť od konkrétneho projektu každého tímu.

Naopak *rozdelenie zodpovednosti* medzi členov tímu bude pre nás podstatné. Každý má svoju funkciu a zodpovedá za istú časť projektu, čo umožňuje efektívnejšiu prácu, ale aj jednoduchšiu kontrolu, kto čo vykonal.

Personálne problémy v zmysle určenia správneho počtu zamestnancov a ich funkcií určite nebudeme riešiť v školskom projekte, nakoľko máme pevne určený počet členov v tíme. V praxi to ale podľa môjho názoru môže byť častý úkaz. Príliš málo zamestnancov značne predĺži celý projekt, môžu nastať problémy pri testovaní systému a podobne. Naopak nadmerný počet ľudí zvyšuje nároky na komunikáciu a koordináciu.

Formálny a pravidelný *odhad rizík* je taktiež užitočný. Z môjho pohľadu, ako som už spomínal, člen tímu zistí riziko a oboznámi s ním manažéra rizík. Spolu ho analyzujú a určia mu prioritu podľa závažnosti. Takto sa získa zoznam identifikovaných rizík zoradený podľa priorít a tie, ktoré sú najvyššie, sa začnú riešiť.

Ďalšou dobrou zásadou je rozdelenie projektu na *malé časti* a ich vypracovávanie v nejakom časovom úseku. Presne toto nás čaká v tímovom projekte, keďže budeme používať agilnú metodiku vývoja. Určíme si úlohy na nasledujúci týždeň pre každého člena a potom si zreferujeme, či sa ich podarilo splniť a zadáme si ďalšie úlohy.

Posledné dve zásady, ktoré spomeniem súvisia s tým, čo som už myslím dostatočne opísal aj v predošlej časti. Ide o *stabilizovanie požiadaviek* a špecifikácie čo najskôr na začiatku projektu. Tak aby sa nemuselo nič prerábať. V prípade, že sa systém člení na menšie podsystémy, je možné aj ich prípadné neskoršie definovanie, lebo tím je uzrozumený, že má vyvíjať produkt tak, aby sa tieto podsystémy dali ľahko „prilepiť“ k celku.

Tou druhou zásadou je implementovanie *primeraného počtu nových funkcií*. Opäť spomeniem „pozlacovanie systému“, aj keď toto má trochu širší význam. V prípade, že plánujeme priniesť novú funkcionality, je vhodné dávkovať ju postupne v po sebe nasledujúcich verziách systému, aby používatelia mali viac času na ich spoznanie. Takto ich príliš nezahltíme, čo znižuje možnosť neprijatia systému, no taktiež máme podľa mňa postarané o dobrú reklamu, lebo pravidelne prinášame niečo nové, snáď aj naozaj prínosné.

Zhrnutie

Na záver by som už len zopakoval to, čo sa mi zdá najhlavnejšie. Na začiatku si projekt dobre naplánovať, špecifikovať požiadavky čo najskôr a čo najpresnejšie. Vytvoriť štruktúru riadenia a vývoja takú, aby sa riziká ľahšie identifikovali, a teda aj spracovávali. Snažiť sa systém udržiavať jednoduchý a prehľadný, odolať pokušeniu pridávať za chodu stále nové funkcie. No a byť pripravený reagovať na vzniknuté situácie. Myslím, že potom by projekt mohol dôjsť do úspešného konca, aj keď menšie problémy sa vyskytnúť môžu.

Použitá literatúra

1. Addison, T., Vallabh, S.: Controlling Software Project Risks - an Empirical Study of Methods used by Experienced Project Managers. *Proceedings of SIACSIT*, 2002, 128-140.
2. Wallace, L., Keil, M.: Software project risks and their effects on outcomes. *Communications of the ACM*, Vol. 47, No. 4 (2004), 68-73.
3. Warkentin, M., Moore, R.S., Bekkering, E., Johnston, A.C.: Analysis of Systems Development Project Risks: An Integrative Framework. *The DATA BASE for Advances in Information Systems*, Vol. 40, No. 2 (2009), 8-27.

Annotation

Risk management in software project

Almost every activity is linked with some risk. Software or any other project is not an exception, of course. This risk may be variously huge and may have various consequences from tiny obstacles to failure of the whole effort made. On the other hand, risk means a challenge which may or may not bring benefit. Therefore it is needed to know about the risks that might come during the work on the project and try to minimize them or find an ideal boundary between threatening the project and gaining some kind of advantage. In this essay, I write about common risk factors and aim for the most important of them in view of our school team project. I also describe the ways of decreasing and eliminating these risks.