

AKO ČELIŤ NÁSTRAHÁM V MENŠÍCH PROJEKTOCH

Aj malý ľadovec dokáže potopiť Titanic.

Vojtech Villaris

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

Abstrakt. *Vývoj softvérových projektov je často spätý s rôznymi vonkajšími a vnútornými rizikami, ktoré môžu spôsobiť rozličné problémy. Existuje viacero publikácií, ktoré poskytujú rozličné návody a techniky na to, ako sa s riadením takýchto rizík vysporiadať, ale drvivá väčšina z nich je určená pre veľké firmy a rozsiahle softvérové projekty. V tejto eseji sa snažím zamerať predovšetkým na menšie projekty, na ktorých pracujú menej členné tímy a tým oblasť riadenia rizík preniesť aj do prostredia semestrálnych projektov riešených v tímoch. Za týmto účelom v eseji opisujem hlavné rozdiely medzi veľkými a malými projektmi a z toho vyplývajúce aj rôzne riziká, ktorým v nich môžeme čeliť.*

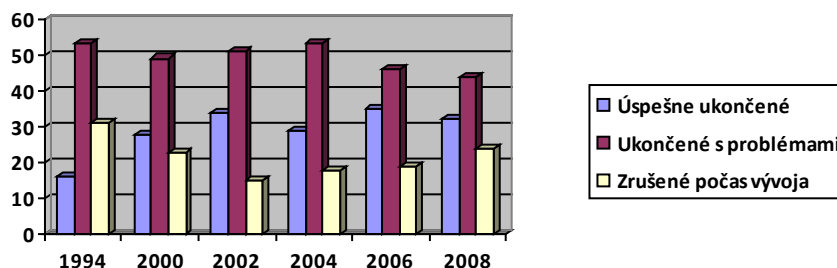
Kľúčové slová: *riziko, manažment rizík, malé tímy, menšie projekty*

Úvod

Riziko je definované ako pravdepodobnosť zlyhania dosiahnutia určitého cieľa, výkonu, naplánovaných vecí a dôsledok zlyhania dosiahnutia tohto cieľa [1]. V dôsledku tohto zlyhania môže byť potom výsledný produkt považovaný z viacerých hľadísk za neúspešný. Čas strávený nad vyvíjaním takéhoto softvérového projektu ako aj financie vynaložené na jeho uskutočnenie môžu prekročiť tie plánované, a čo je horšie, tak výsledný produkt nemusí spĺňať pôvodné požiadavky. Takéto oneskorenie dodania softvéru alebo jeho nefunkčnosť majú zlý vplyv na meno firmy, ktorá ho vytvorila. Je teda zrejmé, že takýmto nepríjemnostiam sa chce každý vyhnúť. Preto v manažmente projektov vzniklo odvetvie manažmentu rizík, ktoré má za úlohu snahu takýmto rizikám predchádzať.

2 *Vojtech Villarís*

To, že manažovanie rizík pri vývoji softvéru je naozaj potrebné, vyplýva aj zo správ The Standish Group [4], ktoré od roku 1994 sledujú úspešnosť vývoja softvéru. Ich výsledky sú znázornené na obrázku Obr. 1, ktorý zobrazuje percentuálnu úspešnosť pri vývoji softvérových projektov v USA v rokoch 1994 a 2000 až 2008. Z tohto porovnania jasne vyplýva, že i keď sa počet úspešne ukončených projektov oproti výsledkom z roku 1994 značne zvýšil, stále existuje výrazný priestor na ďalšie zlepšovanie.



Obr. 1. Percentuálne zobrazenie úspešnosti vývoja projektov v USA.

Vysoké počty projektov, ktoré sú ukončované s problémami, t.j. napríklad po prekročení rozpočtu alebo termínu dodania, môžu naznačovať aj problémy projektov pri odhadovaní potrebných zdrojov a celkových nákladov v úvode ich riešenia. Ak by sme teda vedeli už v tejto fáze v každom projekte určiť problémy, ktorým môžeme počas životného cyklu jeho vývoja čeliť, bolo by možné sa na ne pripraviť, niektoré dokonca úplne eliminovať, a tým teda zvýšiť šancu, že náš projekt skončí úspešne.

Menšie tímové projekty

Na manažovanie rizík vo veľkých firmách a v rozsiahlych projektoch je napísané množstvo publikácií a návodov. Podobná podpora, ktorá by sa dala aplikovať aj na malé projekty však až tak dostupná nie je. Aby sme vedeli zvoliť, ktoré postupy by boli vhodné aj pre takéto menšie projekty, je potrebné si najprv uvedomiť rozdiely, ktoré medzi nimi existujú. Už na prvý pohľad je jasné, že menšie projekty majú menšie rozpočty, menej času na vývoj a pracuje na nich menej pracovníkov. Podľa Donny L. Johnson [3] by sa menšie projekty dali charakterizovať vlastnosťami v tabuľke Tab. 1, v ktorej sú naznačené aj ich vplyvy na manažment rizík.

Tab. 1. Charakteristika menších tímových projektov.

Charakteristika projektu	Vplyv na manažment rizík
Krátky čas vývoja	Nedostatočný čas na vysporiadanie sa s rizikami, Menej možností na ich výskyt.
Obmedzené zdroje	Obmedzený počet ľudí na manažovanie rizík, Nedostatočné zdroje na prijatie dodatočného personálu.
Malé tímy	Rozhodovanie členov tímu o rizikách.
Málo prepojení s okolím	Menej rizík kvôli menšiemu počtu závislostí.

Hlavnou charakteristikou malého projektu bývajú teda obmedzené zdroje. Ako dôsledok si pri tvorbe takýchto projektov nemôžeme dovoliť najímať špecializovaných odborníkov na každú oblasť, ale všetko musí vyriešiť menší počet pracovníkov. Výsledkom potom býva to, že každý zastáva v tíme viacero funkcií a v tom najhoršom prípade sa ani jednej z nich nedokáže venovať dostatočne. S nedostatkom zdrojov súvisí aj relatívne krátky čas na vývoj. Je zrejmé, že menšie projekty budú mať na svoj vývoj menej času ako tie rozsiahle. Z toho vyplýva na jednej strane menej času na každý krok vývoja, a teda aj na samotný manažment rizík, ale na strane druhej zároveň aj menej príležitostí, aby vôbec niektoré riziká mohli nastať.

Z týchto dôvodov sa potom projektovým manažérom môže núkať myšlienka proces vývoja okresať tak, aby ostalo viac zdrojov na samotný vývoj. Veď keď sa nad tým trochu zamyslíme, tak riziká v podstate vôbec nastať nemusia a ich manažment vlastne ani nemá kto urobiť, keďže si odborníka projekt dovoliť nemôže. Potom to môže dopadnúť tak, že keď sa nejaké riziko predsa len objaví, tak jeho riešenie ostane na pleciach projektového manažéra, ktorý pod tlakom nemusí urobiť správne rozhodnutia a projekt tak môže ešte viac ohroziť. Na to, aby sme sa takýmto situáciám vyhli, je podľa mňa vždy lepšie určitý manažment rizík pred projektom vykonať. Vždy je lepšie spraviť aspoň niečo, aby sme sa na budúce problémy pripravili, aj keby to malo byť iba formou diskusie a nápadov vybraných členov tímu. Z tohto dôvodu by bolo pre takéto tímy vhodné vedieť, ktoré riziká sú pre ne aktuálne, a teda na ktoré by sa mali predovšetkým zamerať.

Riziká v menších tímových projektoch

Ako už bolo spomenuté riziká v menších projektoch sa môžu líšiť od tých identifikovaných vo väčších projektoch. Podľa výskumu David A. K. Crosby [2], na ktorom sa zúčastnilo osem softvérových organizácií s počtom zamestnancov od päť do desať, patria medzi najčastejšie identifikované riziká, ktorými sa treba zaoberať tieto:

- nejasná alebo zle pochopená špecifikácia,
- zavádzanie novej technológie,
- nereálny časový rozvrh a rozpočet,
- zlyhanie pri snahe zainteresovať používateľa.

Na základe skúseností z tímového projektu, ale aj ďalších semestrálnych projektov, na ktorých sa pracuje v tíme, by som k nim pridal ešte riziko nedostatočnej motivácie člena tímu na prácu alebo v horšom prípade aj jeho možné opustenie tímu.

Nejasná alebo zle pochopená špecifikácia

Toto riziko je úzko späté so schopnosťou zadávateľa úlohy opísať svoje požiadavky a softvérovej organizácie tieto požiadavky správne pochopiť. S týmto problémom sa v súčasnosti stretávame vo väčšine zadaných zákaziek. Príčina je veľmi jednoduchá. Zákazník totiž vo väčšine prípadov ani sám nevie, čo vlastne chce, a keď to náhodou vie, nemusí to vedieť presne vysvetliť.

Riešenia ako sa následkom tohto rizika v čo najväčšej možnej miere vyhnúť sú v podstate dve. To prvé je založené na komunikácii medzi zadávateľom a vývojárskym

4 *Vojtech Villarís*

tímom. Je veľmi dôležité, aby sa jednotlivé požiadavky na produkt diskutovali do takej miery, dokým nenastane ich vzájomné pochopenie. Veľmi dôležité je aj to, aby takto rozpisované požiadavky v špecifikácii zadávateľ aj potvrdil, aby sa nám neskôr nestalo, že si to zákazník medzičasom rozmyslí a bude očakávať niečo iné ako prvotne chcel. Zároveň by sa mali konzultovať aj významné kroky a rozhodnutia, ktoré môžu mať za následok ovplyvnenie konečnej funkčnosti produktu. Týmto spôsobom si tzv. kryjeme chrbát a zároveň sa môžeme vyhnúť vývoju funkcionality, ktorá nie je predmetom požiadaviek.

Druhou možnosťou ako z nesprávne pochopených požiadaviek vykorčuľovať, je snaha presvedčiť klienta, že to, čo mu ideme odovzdať, je práve to, čo on chcel. Toto riešenie je síce pekné, ale v reálnych softvérových projektoch to môže vyjsť iba zriedkakedy. Záleží tu predovšetkým od odchýlky funkcionality od tej, ktorú klient očakáva. Ak je táto odchýlka malá a väčšinu požadovaných vlastností projekt spĺňa, tak je šanca, že klienta presvedčíme. Ak sme ale zle pochopili nejakú základnú požiadavku, o ktorej sa celý projekt točil, tak je takáto snaha zbytočná ba dokonca až nežiaduca.

Keď zoberieme do úvahy prostredie školy a školských projektov, tak sa tu objavuje aj tento problém. Často sa stáva, že študent aj napriek dobrej práci nedostane za projekt vysoký počet bodov, pretože neurobil to, čo bolo predmetom jej zadania. Veľmi jednoduchým príkladom môže byť aj diplomová práca. Pri jej riešení je veľmi dôležité, aby sa študent držal svojho zadania. Je potrebné, aby práca obsahovala presne to, o čom zadanie hovorí a nedodržanie tejto požiadavky môže vyústiť až do neúspešného absolvovania predmetu. Študentovi vtedy nepomôže ani to, ak by samotná práca bola výborná.

Zavádzanie novej technológie

Na riešenie tohto rizika sa používajú zväčša tréningy [2]. Táto technika spočíva v rôznych školiaciach kurzoch, ktoré pripravujú zamestnancov na danú novú technológiu. Pravdou však je, že keď hovoríme o menších tímoch a menších projektoch, tak sa skôr dáva prednosť „nútenému“ samoštúdiu. Dôvod je jednoduchý – financie. Takéto kurzy totiž nie sú zadarmo. Pomocou však môže byť, ak danú technológiu ovláda aspoň jeden z tímu a vie ostatných na začiatku nasmerovať na tie najdôležitejšie možnosti, ktoré ponúka a zmeny, ktoré sú s jej používaním späté. Pokiaľ je však daná technológia nová pre celý tím, jej zavedenie môže v konečnom dôsledku spôsobiť viac škody ako úžitku.

Prvým problémom je čas. Štúdium novej technológie si vyžaduje čas, ktorý by mohol byť inak investovaný do vývoja. Ak navyše nikto nevie, ako sa presne používa, a čo z nej vlastne chceme použiť, tak si to vyžaduje najprv akúsi analýzu s prihliadnutím na možnosti a obmedzenia tejto technológie a na zadané požiadavky. Ak by sme tento krok nevykonali, tak by sa nám mohlo nakoniec pri implementácii stať, že zistíme, že to kvôli čomu sme sa rozhodli pre použitie novej technológie v podstate nie je to, čo naozaj potrebujeme. Alebo zistíme, že naša nová technológia nám neumožní vykonávať nejakú požiadavku alebo jej časť, pretože to jednoducho nebude podporovať.

So zavádzaním nových technológií sa stretávame aj na akademickej pôde v rámci niektorých predmetov. Keď hovoríme o práci v malých tímoch, tak do úvahy pripadá príklad z tímového projektu. Novú technológiu tu pre náš tím predstavoval rámec Spring. Našou výhodou jeho zavádzania bolo, že jeden člen tímu mal s týmto rámcom dostatok

skúseností a od začiatku si vedel predstaviť, kde sa bude dať rozumným spôsobom použiť. Pokiaľ by nám ale nedal nejaké úvodné školenie a neporadil nám na čo presne sa máme zamerať, tak by jeho zavedenie nebolo až také „bezbolestné“. Viem si dokonca predstaviť aj to, že by sme sa do toho nakoniec vôbec nepustili.

Nereálny časový rozvrh a rozpočet

Toto riziko patrí medzi tie najviac sledované. Je to dané tým, že v súčasnej dobe býva pri projektoch presne určené, kedy sa má hotový produkt odovzdať a za koľko. Tento problém väčšina organizácií rieši podrobnými rozpismi a plánmi práce [2]. V týchto rozvrhoch však nemáme ako počítať s nepredvídanými udalosťami, ktoré môžu zapríčiniť ich prehodnotenie alebo aj zmenenie. Aj keď tieto rozvrhy zväčša nebývajú úplne dodržané, pomáhajú aspoň predchádzať fatálnym odchýlkam od plánu.

Osobne si myslím, že problém s dodržiavaním termínov má väčšina ľudí vo všetkých oblastiach. Je to dané ich pohodlnosťou a tendenciou všetko si nechávať na poslednú chvíľu. Ja osobne tiež patríam medzi túto skupinu a už ma to po pravde stálo aj niekoľko bodov za neskoré odovzdávania. Keby som bol ale schopný vytvoriť si nejaký časový rozpis, podľa ktorého by som nakoniec aj zadávané projekty riešil, tak by som sa takýmto problémom mohol vyhnúť.

Zlyhanie pri snahe zainteresovať používateľa

Ako riešenie tohto problému sa uvádza predovšetkým spätná väzba [2]. Základom je identifikovať jednu alebo zopár hlavných postáv na strane zákazníckej organizácie a tieto postavy sa snažiť do vývoja čo najviac zapájať. V spolupráci s nimi sa vytvárajú aj akceptačné testy, ktorých úlohou je uistiť sa, že produkt spĺňa funkčné požiadavky ešte pred jeho samotným odovzdaním.

Toto riziko sa v projektoch bežne riešených počas štúdia v podstate nevyskytuje. Je to dané tým, že väčšina projektov sú samoučelné aplikácie, ktoré slúžia na vyriešenie určitého zadaného problému a používateľom v nich je sám študent alebo, ako to bolo v prípade nášho tímového projektu, vedúci projektu, keďže aplikácia bola určená pre profesorov. Z tohto dôvodu by som riziko zlyhania zainteresovania používateľa, z hľadiska výskytu v semestrálnych projektoch, nebral ako veľmi významné.

Nedostatočná motivácia

Následkami nedostatočnej motivácie člena tímu na zapájanie sa do riešenia projektu trpí zväčša celý tím. Uvedomenie si tohto rizika je preto mimoriadne potrebné v každom projekte, na ktorom pracuje viacero ľudí. Pokiaľ sa nie je možné spoľahnúť na to, že každý svoju prácu odvedie v čas a na dostatočnej úrovni, tak sa môže projekt dostať veľmi rýchlo do problémov. Na nedostatočnú motiváciu v akademickej sfére môže vplývať viacero faktorov. Tými najzákladnejšími sú lenivosť a pohodlnosť ľudí. Nie každý totiž chce mať z každého predmetu výborné hodnotenie a niekomu stačí, aby predmet ukončil s čo najmenšou námahou. Ak sa nám takéhoto člena tímu nepodarí presvedčiť aby dostatočne spolupracoval, treba sa s tým zmieriť a prispôbiť tomu rozdeľovanie úloh tak, aby spravil aspoň tie, ktoré mu budú pridelené.

Odchod člena tímu

Pod týmto rizikom sa skrývajú rôzne príčiny, prečo určitý člen tímu odíde. Vo všeobecnosti nimi môžu byť napríklad vzájomné nezhody, nedostatočné vedomosti, zdravotný stav alebo môžu byť dôsledkom predchádzajúceho rizika, čiže nedostatočnej motivácie na riešenie projektu. Následky, ktoré vyplývajú z odchodu člena tímu závisia aj od typu daného tímu. Je rozdiel, či je tím vytvorený v rámci väčšej spoločnosti pridelením zamestnancov na menší projekt, kde sa dajú zamestnanci v prípade potreby nahradiť alebo si vybrať nemôžeme a všetci členovia tímu sú vopred daní. Ak nastane odchod z uzavretého tímu, aké predstavujú napríklad aj tímy na tímových projektoch, tak to bude znamenať viac práce pre zvyšok tímu, ktorý bude musieť prehodnotiť svoje roly v tíme a prerozdeliť aj tie, ktoré mal odchádzajúci člen.

Pokiaľ odchod nastane už v rozbehnutom projekte, môže nastať situácia, že tím nebude vedieť, ako ďalej pokračovať v úlohách, na ktorých chýbajúci člen tímu pracoval sám. Z tohto dôvodu sa odporúča, aby sa na zložitejších úlohách pracovalo aspoň vo dvojici a zároveň bolo každé rozhodnutie ovplyvňujúce projekt v dostatočnej miere komentované. Ak sa budeme riadiť týmito zásadami, tak aj keby sa stalo, že niekto tím opustí, vždy ostane niekto ďalší, kto bude vedieť v jeho práci pokračovať.

Záver

V eseji som rozobral hlavné rozdiely medzi projektmi vyvíjanými malými a veľkými tímami s dôrazom na dôsledky na možnosti vykonávania manažmentu rizík. Vzhľadom na to, že komplexný manažment rizík si niektoré takéto menšie tímy nemôžu dovoliť vykonávať, či už vzhľadom na jeho časovú náročnosť alebo nedostatočné vedomosti z oblasti manažmentu rizík, som sa pokúsil identifikovať iba tie najdôležitejšie z nich, ktorým môžu čeliť. Tieto riziká som ďalej priblížil práve z pohľadu menších tímov, ktoré sa vyskytujú aj pri riešení semestrálnych projektov na akademickej pôde a načrtol som aj ich možné riešenia.

Použitá literatúra

1. Conrow, E.H., Shishido, P.S.: , Implementing risk management on software intensive projects. *Software, IEEE* , Vol.14, No. 3 (1997), 83-89.
2. Crosby, D.A.K.: *Project Risk Management In Smaller Software Teams* [online]. 2007. [cit. 02.10.2010]. Dostupné na:
<http://repositoryaut.lconz.ac.nz/bitstream/10292/378/3/CrosbyD.pdf>
3. Johnson D.L.: *Risk Management and the Small Software Project Risk* [online]. 2006. [cit. 05.10.2012]. Dostupné na:
<http://www.sei.cmu.edu/iprc/sepg2006/johnson.pdf>
4. The Standish Group International. *Chaos Summary for 2010* [online]. 2010. Boston, USA. [cit. 02.10.2012]. Dostupné na:
<http://insyght.com.au/special/2010CHAOSSummary.pdf>

Annotation

How to face the risks in smaller projects

Development of software projects is often linked with different outer or inner risks, which might cause various problems. There are several publications providing some guidelines and techniques for mitigating such risks but most of them are designed for large companies and are focused on bigger software projects. For this reason my essay is primary oriented toward smaller teams that are developing small scale projects. Its aim is also to transpose the process of managing risks to academical ground of semestral projects developed in teams. Therefore this essay describes main differences between large and small projects as well as risks that are considered the greatest ones in this area.