

AKO SI NEZLOMIŤ RUKU ?

Prvý krát si za svoje štúdium uvedomujem riziko

Alojz Gomola

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

Abstrakt. Harry W. Boehm identifikoval desať najväčších rizík ešte v roku 1989, tieto riziká ostávajú aktuálne do súčasnosti. Podmienky a prostredie malého projektu sú odlišné, ľudia nie sú tak silno zviazaní hierarchiou moci a ich motivácia nie je závislá na peniazoch. Odstránenie možných rizík je životne dôležité pre úspešnosť každého projektu. Esej prináša pohľad na tieto riziká v prostredí malého projektu. Pokúsime sa identifikovať, ktoré riziká sú najnebezpečnejšie, porovnáme rôzne metódy identifikácie týchto rizík. Sústredíme sa najmä na ľudský faktor a s ním spojené riziká. Analýzu daných rizík premietneme do riadenia projektu. Rozoberieme vplyv rizika na plánovanie, porovnáme klasický prístup k rozloženiu rizika s optimistickým plánovaním, zhodnotíme dopad oboch metód na plán projektu.

Kľúčové slová: manažment rizík, malý tím, konflikty, optimistický plán, motivácia

Úvod

Vzhľadom na to, ako dynamicky sa vyvíja prostredie informačných technológií, je nepochopiteľné ako pri takom množstve nástrojov na správu projektu nie sme schopní úplne eliminovať riziká s ním spojené. Zoberme si ako príklad Spojené štáty americké, ktoré majú na poli softvérových systémov vedúcu pozíciu. Aj veľkí hráči ako IBM, SAP, McDonald mali niekoľko neúspešných softvérových projektov. Veľmi ťažko sa uverí aj fakt [9] že 70% projektov má problémy, či už to sú zvýšené náklady alebo oneskorené odovzdávanie výstupov. Toto všetko je spôsobené neošetrením zdrojov rizika v daných projektoch [5], asi 20 % projektov sa nepodarilo dokončiť vôbec. Pre lepšiu predstavu: čisté

náklady na tieto projekty boli „iba“ 59 miliárd dolárov. Keď si zoberieme, že tieto čísla reflektujú stav spred niekoľkých rokov, uvedomíme si dôležitosť správneho riadenia.

Pre veľké firmy nebýva problém zniesť náklady za *zabraný projekt*, pre malé organizácie, ktoré doslovne žijú od projektu k projektu to *má fatálne následky*. Pre nás je podstatné to, že daná situácia nie je neriešiteľná. Za určitú cenu sa môžeme niektorým situáciám vyhnúť. Keďže aj malé projekty majú čím ďalej tým väčšiu zložitosť, správne aplikovaný manažment rizík naberá patričnú dôležitosť aj v tejto doméne. Mnohé štúdie sa zaoberali manažmentom rizík v oblasti veľkých softvérových projektov [4,5,8], ktoré zahŕňali niekoľko stovák pracovníkov, odhalili riziká v takýchto projektoch, ktoré korešpondovali s *Boehmovým zoznamom*. Keďže viacero odborníkov z oblasti manažmentu rizík považuje tento zoznam za *stále aktuálny* [7], použijeme ho aj mi ako základ pre naše ďalšie bádanie. Postupne rozoberieme jednotlivé riziká z pohľadu malého softvérového projektu, na ktorý aplikujeme špecifické obmedzenia prostredia fakulty.

Cieľom eseje je *preskúmať tieto riziká* a priniesť ucelený pohľad na ich *vplyv na celkovú úspešnosť projektu*, bližšie sa pozrieme najmä na riziká spojené s ľuďmi, a ich vplyv na plánovanie softvérového projektu

Čo vlastne považujeme za riziko ?

Riziko je udalosť, ktorá keď nastane, spôsobí na výstupe [9] nevýhodu alebo stratu materiálnu/nemateriálnu.

Riziko môžeme špecifikovať jeho zložkami [8]:

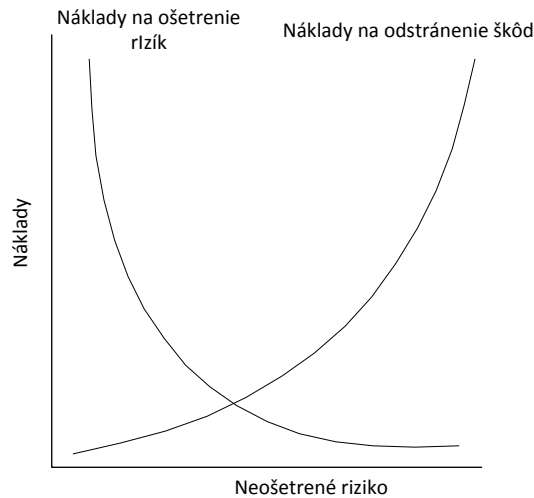
1. Zdroj
2. Pravdepodobnosť

Zatiaľ čo *zdroj rizika nemôžeme úplne odstrániť*. V niektorých prípadoch, môžeme zmenšiť pravdepodobnosť výskytu rizikovej situácie. Zníženie pravdepodobnosti nás môže stať ďalšie prostriedky, ale je všeobecne platné, že čím skôr odstránime identifikované riziko, tým viac ušetríme na celkových nákladoch [9].

Ako príklad môžeme uviesť riziko požiaru v dátovom úložisku a s ním spojené riziko straty dát. Máme viacero možností ako eliminovať dané riziko, buď znížime pravdepodobnosť požiaru, alebo vytvoríme záložné dátové úložisko. Ani jedna možnosť neeliminuje dané riziko, ale obidve značne znižujú pravdepodobnosť nastania straty dát.

Vplyv rizika na celkové náklady

Náklady spojené s ošetrením rizika sú rádovo menšie ako náklady spojené s odstránením škôd [8]. Dôležité však je, ktoré riziká sa rozhodneme odstrániť, pretože do celkových nákladov musíme počítať aj náklady spojené s ich odstránením. Štúdia [9] vyvodzuje dôsledok, že vhodné je odstrániť určité percento identifikovaných rizík, a to tak aby sme dosiahli optimálne predpokladané náklady spojené s neošetreným rizikom. Tento fakt môžeme názorne ilustrovať pomocou (Obr. 1.). Ako máme možnosť vidieť, ideálny stav nastane, keď sa s neošetrenými rizikami nachádzame na priesečníku nákladových kriviek.



Obr. 1. Vplyv neošetreného rizika na náklady.

Za zdroje pokladáme čas vynaložený na projekt aj prostriedky vynaložené na projekt, keďže v prostredí tímového projektu za zdroj pokladáme len čas, snažíme sa minimalizovať čas vynaložený na dosiahnutie cieľa. Naším cieľom je dosiahnuť pri minimálnom úsilí maximálny možný výsledok. Toto sa nám často nedarí naplniť vzhľadom na nedostatok skúseností, častokrát zlyhávame na slabom riadení zdrojov.

Ako vlastne manažovať riziká ?

Vieme, čo je to riziko, vieme ako môže vplývať na náš projekt, ale ako s ním pracovať? Mali by sme sa naučiť riziko *identifikovať a minimalizovať pravdepodobnosť jeho vzniku*. Najprv si však špecifikujeme proces manažmentu rizík a prístup k jeho vykonávaniu. Z pohľadu životného cyklu softvérového systému (SDLC) na:

- *lokálne* – vyskytujúce sa len v určitej fáze SDLC
- *globálne* – vyskytujúce sa počas celého SDLC

Z pohľadu projektu:

- *interné* – vyskytujúce sa vo vnútri organizácie
- *externé* – vyskytujúce sa mimo organizácie (napr. živelná pohroma)

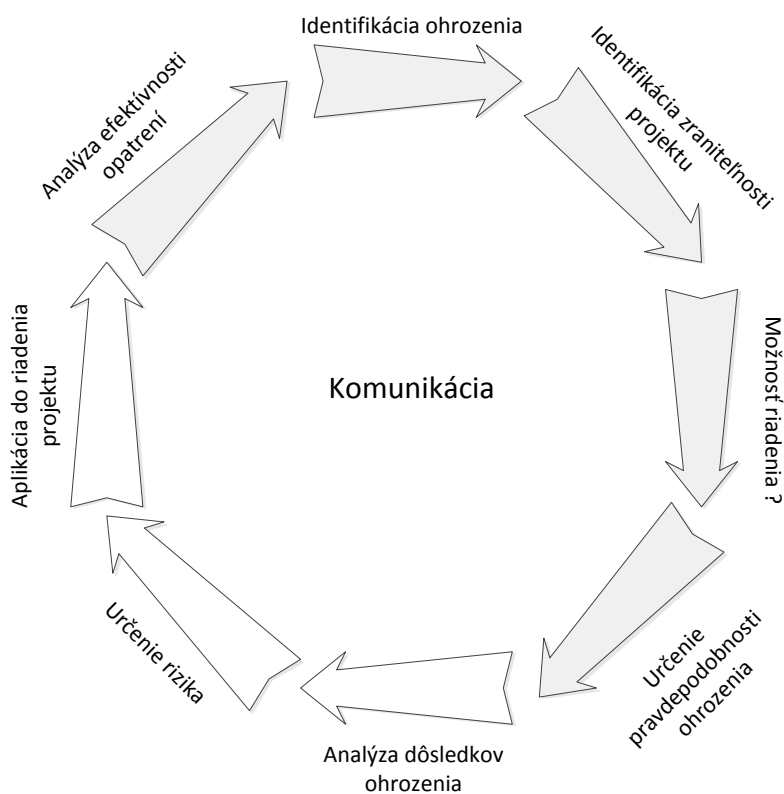
Pre niektoré riziká nám postačuje vodopádový model manažmentu rizík. Teda: identifikujeme ich, minimalizujeme ich dopad a aplikujeme na celý projekt, ale to nie je postačujúce.

Podľa [1] by mal byť prístup k manažmentu rizík iteratívno-inkrementálny lebo:

- vznikajú rôzne riziká počas jednotlivých fáz životného cyklu softvérového systému (SDLC)
- mení sa pravdepodobnosť vzniku rizikových udalostí aj ich celkový dopad na projekt

Celkovo prístup k manažmentu rizík [8] vyjadríme ako rozšírený Sheward/Demingov kruh (pozri Obr. 2) pozostávajúci z nasledovných činností:

1. *Identifikácia ohrozenia* – identifikujeme možné zdroje ohrozenia pre náš projekt
2. *Identifikácia zraniteľnosti projektu* – identifikujeme ako môžu zistené ohrozenia ovplyvniť celkové fungovanie projektu
3. *Určenie možností riadenia ohrozenia a ich nákladovosť* – načrtujeme predbežné riešenia na identifikované ohrozenia
4. *Určenie pravdepodobnosti ohrozenia* – odhadneme pravdepodobnosť nastania ohrozenia
5. *Analýza dôsledkov ohrozenia* – analyzujeme dôsledky ošetrenia/ neošetrenia daného rizika v projekte a určíme možné dopady
6. *Určenie rizika* – určíme závažnosť rizika pre daný projekt a vypracujeme postup jeho riadenia
7. *Aplikácia do riadenia projektu*– aplikujeme získaný postup do riadenia projektu
8. *Analýza efektívnosti opatrení* – overíme efektívnosť daného postupu, ktorý sme aplikovali, vyvodíme dôsledky



Obr. 2 Proces manažmentu rizík.

Daný postup je efektívny iba keď funguje komunikácia medzi jednotlivými oddeleniami organizácie [4]. V našich podmienkach môžeme toto tvrdenie pretransformovať nasledovne, že daný

postup je efektívny iba keď funguje komunikácia na úrovni tímu. Situácia je však sťažená nedostatočnou autoritou jednotlivca v danom kolektíve, preto je potrebná spolupráca s vedúcim pedagógom (vedúci oddelenia) [4], ktorý pomáha aplikovať manažment rizík na globálnej úrovni.

Nový pohľad na Boehmov zoznam

Ako sme už v abstrakte spomínali *Boehmov zoznam* je aktuálny dodnes [5]. Jeho kategorizácia rizík je pomerne dobrá, ale niektoré by sa dali zlúčiť do skupín, aby sme ich mohli lepšie spracovávať. Vychádzal som zo štúdie [7] v ktorej pôvodných desať rizík zlúčili do *šiestich komponentov rizika*. Táto štúdia potom pomerne dobre spracovávala jednotlivé komponenty a ich dopad na softvérový projekt. Štúdia bola zameraná na veľké projekty, ale niektoré zistenia sú platné aj v malých projektoch. Vzťahy medzi jednotlivými rizikami a komponentmi reflektuje (Tab. 1).

Tab. 1 Riziká podľa Boehma a rozloženie komponentov rizika.

Riziko	Komponent
Nedostatok personálu	Riziká správy personálu
Nerealistické rozvrhy a rozpočty	Plánovanie a načasovanie
Vývoj zlej funkcionality	Systémová funkcionality
Vývoj zlého používateľského rozhrania	
Pozlacovanie systému	Manažment požiadaviek
Neustále zmeny požiadaviek	
Nedostatky v externe vytvorených moduloch	Externe dodávané komponenty
Nedostatky v externe zabezpečených úlohách	
Nedostatočný výkon v reálnom čase	Využívanie zdrojov a požadovaný výkon
Precenenie technológie	

Špecifiká malého tímu

Čo sa týka tímového projektu, sme *silno ohraničení*, či už z hľadiska *technického zabezpečenia* tímu, či už *zloženia tímu*. Väčšinou to býva tak, a nie je to len môj prípad, že dostaneme, čo sme nechceli, či už sú to ľudia, či už je to téma tímového projektu. S týmito skutočnosťami sa ťažko vyrovnáva a správny manažér ich zohľadní nie len pri plánovaní, ale aj pri riadení projektu. Obrovským problémom je *nedostatok motivácie* v takomto tíme. Ľudia sú často motivovaní len získaním známky a kreditov, správny manažér si je vedomý nasledovnej skutočnosti a zohľadní ju vo svojich odhadoch. *Dobry odhad stavu veci býva základom pre správny plán* [6]. Medzilidská komunikácia je v tíme na najvyššom stupni. Jasné definovanie úloh môže zabezpečiť dobré fungovanie spolupráce, ale nesmieme zabúdať na fakt, že práca nemôže byť rovnomerne distribuovaná, vzhľadom na skúsenosti jednotlivca s danou technológiou.

Ďalším podstatným problémom na, ktorý sme narazili, je *faktor ega jednotlivca* a jeho vplyv na celkové fungovanie tímu. Agilné metódy vývoja a „*egoless programming*“ by to síce mohli vyriešiť, ale vo väčšine tímov sú ignorované. Veľkým problémom je aj viditeľnosť

výstupu. Stáva sa, že človek, ktorý programuje „back-end“ aplikácie má pocit menejcennosti voči človeku, ktorý programuje „front-end“ a má možnosť vidieť plody svojej práce. Ako príklad môžem uviesť prípad databázistu, ktorý pracoval na hranici svojich možností, ale nevidel výstup svojej práce. Možným riešením tohto rizika by bolo, aby si ľudia v projekte uvedomili svoju závislosť práve na takýchto ľuďoch. Bohužiaľ, drvivá väčšina si toto uvedomí, až keď tá časť, na ktorej sú závislí, prestane fungovať [6].

Ak by sme mali tímový projekt k niečomu prirovnať, tak by to bol taký malý krst ohňom. Síce naša fakulta je známa tým, že väčšina jej študentov pracuje popri škole, ale do manažérskej role sú postavení prvýkrát. Manažment vyžaduje vyššiu mieru abstrakcie ako bohapusté písanie kódu. Čo sa týka manažmentu rizík, tak je dokázané, že lepší manažment rizík vykonávajú ľudia, ktorí majú niekoľko projektov za sebou [5,6].

V mnohých projektoch sa dajú termíny posúvať, takéto prípady však spadajú do kategórie, kde zlyhal manažment. V našom prípade sú *termíny fixné a ich nedodržanie znamená koniec projektu*. Nezáleží na tom, do akej miery bol projekt splnený, toto pôsobí na niektorých členov tímu demotivujúco a zákonite podávajú slabší výkon.

Keď si to zhrnieme naše obmedzenia sú nasledovné:

- slabý rozpočet
- neriešiteľnosť personálnych výpadkov
- slabá motivácia
- zlé odhady v plánovaní
- fixné termíny, ktoré treba dodržiavať
- obmedzené technologické prostriedky
- obmedzené znalosti vývojových a podporných prostriedkov

Z pohľadu týchto obmedzení budeme pristupovať aj k jednotlivým komponentom rizika. Zameriame sa najmä na tie komponenty, ktoré najviac ovplyvňujú výkon teda: personál, plánovanie a požiadavky. Ostatné prejdeme len okrajovo.

Riziká správy personálu

Boehm toto riziko nazval *nedostatok personálu* [3], Lyytinen [7] však pridal ešte *kvalitatívnu zložku personálu*. Záleží aj na tom, akú znalosť technológií majú daní ľudia, aké sú ich povahové charakteristiky. Mohli by sme namietiť, že dnes keď ročne v Indii spromuje dvestotisíc softvérových inžinierov, čosi ako personálne riziká nemôže existovať. Stále však platí, že kvalita je viac ako kvantita a predchádzajúce tvrdenie považujeme za veľmi krátkozraké.

Vybrať najlepších ľudí, ktorí sú navyše aj vysoko motivovaní, by bolo istým riešením. Keď si toto riešenie premietneme do prostredia tímového projektu, zistíme, že je nemožné ho realizovať, lebo *ľudia sú nám pridelení, aj projekt je pridelený*. Retting [6] uvádza, že *omnoho vhodnejšie je ľudí si zaškoliť, respektíve vycvičiť pre potreby projektu*.

Personálne výpadky na neklúčových pozíciách sú vo veľkých projektoch takmer nepostrehnuté. V tímovom projekte je daný výpadok bolestný, keď si uvedomíme, že každý človek je nenahraditeľný. Jeho výkon, sa musí rozdeliť na ostatných členov tímu. Možnou *prevenciou personálneho výpadku je preťažovanie tímu* [6], každému členovi projektu sa pridelí o niečo väčší podiel práce (asi o 1/6-1/7 viac). V prípade výpadku, nie je tento

výpadok až tak citeľný, ako v prípade nárazového zvýšenia záťaže. Podotýkame, že je nutné odhadnúť vhodnú mieru preťaženia jednotlivca, aby nedochádzalo k syndrómu vyhorenia.

Špecializácia je nutná pre fungovanie väčších organizácií, pri malom projekte býva nežiaducim javom. Ako príklad môžeme uviesť, že máme jedného človeka, ktorý sa stará o databázu, nazvime ho Jano. Jano má len svoj piesoček, keď niekto chce niečo s databázou, tak mu Jano napíše dotaz, keď niekto chce ukladať nové údaje, tak mu Jano prerobí dátový model. Všetko funguje a všetci sú v pohode až do dňa, keď si Jano zlomí väz a nik s projektom bez neho nevie pohnúť. Považujeme za potrebné, aby v projekte bola nižšia miera špecializácie, aby bol každý nahraditeľný. Správny *odhad miery špecializácie* je dôležitým komponentom manažmentu rizík [7], aby na každého člena tímu pripadal aspoň jeden člen tímu, ktorý je ho *schopný zastúpiť v prípade jeho výpadku*. Laicky povedané nebudeme mať ľudí čisto na dokumentáciu ani čisto na kódovanie.

Keď si to zhrnieme dobrý manažér by mal svoj *tím preťažovať, aby ho pripravoval na prípadnú stratu* a takisto by mal dbať na *fluktuáciu ľudí na jednotlivých úlohách, aby udržal znalosť jednotlivých činností*.

Plánovanie a načasovanie

Plánovanie zdrojov je dôležité preto, lebo si *nemôžeme dovoliť žiadne oneskorenia*, ako sme už spomínali, oneskorenie odovzdania výstupu je pre nás fatálne a ním končí celé naše snaženie. Mať dobrý *plán s reálnymi odhadmi* je v našom prípade ťažko dosiahnuteľné, vzhľadom na naše veľmi malé skúsenosti s plánovaním, istým riešením by bolo použiť agilné metódy vývoja softvéru ako napríklad SCRUM [6], ale mnoho tímov sa ubera iteratívno–inkrementálnym spôsobom vývoja produktu.

V štúdiu [6] bolo spomenutých viacero postupov pre minimalizáciu rizika pri plánovaní, medzi hlavné odporúčania patrilo vytvorenie čo *najväčšieho počtu kontrolných bodov*. Pri tomto postupe však narážame na problém: ako tieto body určiť? Sme síce nútení týždenne vykazovať činnosť na spoločných stretnutiach, ale nemáme však reálnu možnosť overiť si, ako bola táto činnosť náročná, respektíve koľko daná činnosť zaberala času.

Riziká spojené s plánovaním boli identifikované v štúdiu [7], *toto riziko sa považuje za jedno z najzávažnejších*. Správny odhad pri plánovaní nepovažujú za až taký podstatný, s čím hlboko nesúhlasíme, pretože plánovanie je o realistických odhadoch. Je vhodné, pokiaľ sa plán nedodrží v zmysle skorého vykonania, samozrejme bez straty kvality výstupu. Identifikovali niekoľko faktorov, ktoré ovplyvňujú úspešnosť manažmentu rizík v danom komponente rizika a to sú:

- skúsenosti v oblasti plánovania
- zohľadnenie metodiky plánovania pri plánovaní
- veľkosť projektu, ktorý bude riadený
- skúsenosti s riadením obdobných projektov
- použitá technológia

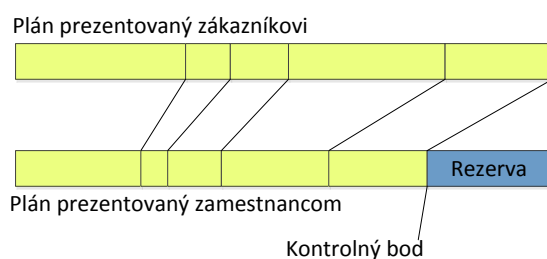
Je dokázané, že projektový manažér [7] je schopnejší lepšie plánovať projekt a zohľadniť v ňom riziká s projektom spojené, ak plánoval aspoň štyri projekty obdobnej veľkosti. Veľkosť projektu ovplyvňuje mieru rizika s ním spojenú, riziko exponenciálne rastie s

veľkosťou projektu. Je väčšia pravdepodobnosť, že *problémy spojené s plánovaním odhalíme skôr ak je projekt menší*. Odporúčanie autorov [7] je *udržať projekt čo najmenší*, ideálne by bolo mať skupinu nezávislých tímov pracujúcich na jednotlivých častiach projektu.

Toto však nemôžeme úplne dosiahnuť, pretože niektoré časti projektu sú na sebe závislé. Preto je dobré pri plánovaní zohľadniť aj časové väzby jednotlivých úloh. Pri zviazaných úlohách vzniká veľké množstvo konfliktov. Najvypuklejší je asi problém medzi kódermi a testerami [4], tieto dve skupiny sú silno zviazané a potrebujú na svoju prácu dostatok času. Navyše sú obojstranne závislí na výstupe svojej práce. Prevencia konfliktu medzi jednotlivými skupinami by mala byť zohľadnená pri manažmente rizík.

Posledným z vymenovaných faktorov je použitá technológia. Veľké organizácie nemajú problém zaškoliť alebo zamestnať nových ľudí, aby pokryli svoj technologický deficit. V malom projekte, kde máme obmedzené možnosti, by sme mali pri vytváraní prvotného plánu zohľadniť aj čas na naučenie sa technológií používaných v projekte. Je lepšie mať človeka, ktorý sa to naučí na začiatku projektu, ako človeka, ktorý sa to bude učiť počas projektu. Nielen, že je ohrozená kvalita jeho výstupov, ale počas celého projektu má nižší výkon. Dobrý manažér rizík si uvedomuje, že náklady na chybu, ktorá sa odstráni na začiatku, sú rádovo menšie ako náklady spojené s jej odstránením na konci.

Informatici žijú v ríši divov, a radi *preceňujú svoje schopnosti*. Sú veľmi leniví a keď, nie sú motivovaní osobným ziskom nesnažia sa. Podľa odborného článku [2] ľudia radi podliehajú javu doťahovania, vedia, že danú úlohu môžu splniť omnoho skôr, ale radšej prácu naťahujú, aby vyšla na termín. Tento jav je opísaný Parkinsonovým zákonom ako: „Konzument sa snaží prispôbiť svoju spotrebu produkcii zdroja.“. Tomuto javu sa nedá zabrániť a je prirodzený. Je tu však určité riešenie ako zamestnanca motivovať k tomu aby pracoval na hranici svojich možností. Vo veľkých firmách radoví zamestnanci nepoznajú zmluvné termíny, ktoré majú dohodnuté k danému projektu. Preto manažment môže zohľadniť riziko pri plánovaní vytvorením optimistického plánu [2] s určitou rezervou (Obr. 3). Štruktúra takéhoto plánu sa nemení, menia sa len termíny jednotlivých kontrolných bodov. Nepovažujeme daný postup za neetický alebo zavádzajúci, len zohľadňuje riziko a zvyšuje výkon personálu.



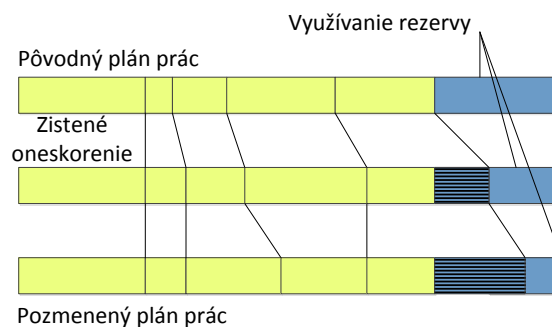
Obr. 3 Dva plány práce.

Niekedy sa stáva, že daný plán je príliš optimistický, preto máme vytvorenú rezervu, aby sme mohli niektoré etapy medzi jednotlivými kontrolnými bodmi naťahovať. Článok [2] uvádza, že vhodné je skrátiť etapu testovania a implementácie, v žiadnom prípade by sme nemali skracovať etapu úvodnej analýzy alebo návrhu. Predchádzajúce tvrdenie je však v ostrom kontraste so štúdiou [4], kde odporúčajú dbať najmä na fázu testovania. V

prostredí malého projektu, by sme sa skôr priklonili k názoru, že testovanie musíme uprednostniť, keďže nemáme takú veľkú používateľskú základňu a odhalenie možných chýb je oneskorené alebo vôbec nenastane. Odhliadnuc od jednotlivých etáp vývoja softvérového systému alebo použitej metodológie vývoja, je vhodné, aby bola vypracovaná metodológia na využívanie rezervy ktorú takýmto plánovaním vytvoríme.

Pri každom očakávanom využití rezervy by sme mali vypracovať zoznam možných posuvov, a dopad na celkový plán projektu. Takéto plánovanie nám značne zmenší mieru rizika oneskorenia dodania jednotlivých výstupov. Pri zvažovaní možného rizika by sme mali brať do úvahy aj koľko času z rezervy pridelieme na riešenie vzniknutých problémov, tento čas by určite nemal prekročiť hodnotu pôvodnej rezervy [2].

Ako príklad môžeme uviesť plán, ktorý je rozložený do piatich etáp (Obr. 4). V prvej etape nevzniknú žiadne oneskorenia oproti plánu prác, v druhej etape príde k oneskoreniu, využijeme značnú časť rezervy, máme väčšie znalosti problémovej oblasti projektu ako na začiatku, preplánujeme nasledujúce etapy. Počas ďalších etáp môže vzniknúť ďalšie oneskorenie oproti plánu, pri každom oneskorení sa pokúsime preplánovať s ohľadom na ostávajúcu rezervu. Ak sa niektoré etapy skrátia tak daný čas pridávame do rezervy.



Obr. 4 Využívanie rezervy.

Využitie rezervy z pohľadu manažmentu rizík považujeme za dôležité, správna manipulácia s časovou rezervou, by mala byť zárukou úspechu daného projektu. Táto manipulácia si však vyžaduje skúsenosti, čo je veľmi ťažké pre začínajúceho manažéra.

Systémová funkcionlita

Lyytinen [7] zlúčil do tohto komponentu dve riziká, vypracovanie nevyhovujúcej funkcionality a z nej vyplývajúce vytvorenie nevyhovujúceho používateľského rozhrania. V prípade vodopádového modelu vývoja softvéru je toto riziko takmer neodstrániteľné, dobre je odhaliteľné pri iteratívno-inkrementálnom vývoji, a asi najlepšie sa s ním dá pracovať pri agilných metódach vývoja [7].

Častá komunikácia zo zákazníkom a najmä prototypovanie rozhraní by malo minimalizovať toto riziko. Problém je však, keď vývojári a zákazník „nehovoria rečou jedného kmeňa“. Považujeme za potrebné, aby sa vývojári dokázali svojim spôsobom myslenia priblížiť zákazníkovi, aspoň pri návrhu funkcionality. Uvedomme si, že používateľské rozhranie a

jeho funkcionalita je jediný reálny výstup, okrem kopy dokumentácie, ktorý od nás zákazník dostáva. Preto nám musí záležať na jeho kvalite aj po estetickej stránke.

Zákazníka by sme už vôbec nemali považovať za dokonalého, zákazník je ako malé dieťa, ktoré nevie povedať akú hračku chce. Napríklad: zákazník chce aby sme mu urobili ďalšie kolónky do formulára, my mu slušne vysvetlíme, že tie údaje vieme vybrať z databázy. Dobrým zvykom býva konzultovať [5] každú navrhovanú zmenu funkcionality so zákazníkom, predchádza sa tým mnohým vynúteným zmenám, ktoré nás v konečnom dôsledku stoja prostriedky.

Dobry manažér rizík by mal čo najviac komunikovať so zákazníkom ohľadom funkcionality, táto časť je úzko prepojená s manažmentom požiadaviek, lebo z požiadaviek častokrát vyplýva navrhovaná funkcionalita [7].

Manažment požiadaviek

Tento komponent rizika je najviac pod kontrolou zákazníka, lebo on zadáva požiadavky. Úlohou manažmentu rizík je zabrániť zásadným zmenám požiadaviek, poprípade špecifikovať niektoré nejasné požiadavky [7]. V malom tíme vystupuje ako zákazník vedúci, ktorý pochádza z prostredia informatikov, čiže vie jasne povedať čo chce, jediným rizikom by mohlo byť, že toho chce veľa. Kvantitu požiadaviek si vieme uregulovať vďaka týždenným stretnutiam.

Externe dodávané komponenty

Externe zabezpečené úlohy a externe vytvorené moduly sú silným komponentom rizika [7], vo všeobecnosti sa táto problematika rieši preskúmaním externe zabezpečenej úlohy. Či už je to formou auditu pri úlohe alebo formou prehliadky a výkonnostných testov pri moduloch. Dobry manažér rizík by mal mať [5] vypracované postupy na ohodnotenie práce externistov, nezáležiac od veľkosti projektu.

Za externe vypracovaný modul môžeme v tímovom projekte považovať minuloročný projekt, na ktorý naviažeme. Je potrebné urobiť dôkladnú analýzu existujúceho riešenia a prípadné úpravy pred pridaním vlastného inkrementu. Takisto sa odporúča prebrať niektoré metodiky, ktoré vyhovujú našim potrebám.

Využívanie zdrojov a požadovaný výkon

„Máme stroje schopné vykonať milióny operácií za sekundu! Tak prečo mi to už tretí deň odosiela jeden formulár?“, aj takto môže vyzeráť práca s viac používateľským systémom. Posledný komponent rizika by sme mohli opísať ako [7] precenenie technológie a s ním spojené nedostatky výkonu v reálnom čase. Často sa stretávame s nepochopením u zákazníka, [5] *zákazník by chcel mať aj esteticky pekný systém aj výpočtovo rýchly systém.* My sme zviazaní prostriedkami na nákup techniky aj na nákup technológie, na ktorej vyvineme produkt.

Spory medzi jednotlivými skupinami vývojárov o to aká technológia sa má použiť by mali byť pod drobnohľadom manažmentu, respektíve *manažment na základe odporúčaní by mal vybrať najvhodnejšiu technológiu pre vývoj daného produktu* [5]. Ako príklad môžem

uviesť, keď bola vybraná technológia zo silnou podporou používateľského rozhrania, ale so slabou podporou databázového systému, výsledok bol taký, že aplikácia síce bola používateľsky príťažlivá, ale zďaleka nie efektívna.

Záver

S narastajúcou zložitou systémov, narastá aj riziko spojené s ich vývojom, preto je manažment rizík dôležitou zložkou manažmentu projektu. Riziko je potrebné regulovať počas celého životného cyklu softvérového systému alebo projektu.

Manažment rizík je spojený proces, ktorý pozostáva z *identifikácie, analýzy, minimalizácie dopadu zdroja ohrozenia*. Dôležité je vypracovanie podrobných postupov s dostatočnou úrovňou abstrakcie a ich následná aplikácia v prípade vzniku ohrozenia.

Oblasť manažmentu rizík je pomerne preskúmaná oblasť a bolo vypracovaných mnoho odborných štúdií. Tieto štúdie poskytujú pohľad na projekty väčšieho rozsahu, odrážajú sa od *Boehmovho zoznamu*.

My sme si zobrali modifikovaný *Boehmov zoznam*, ktorý sme rozdelili na *komponenty rizika*. Dané *komponenty* sme preskúmali v prostredí malého projektu. Opísali sme ich vplyv na projekt a možné perspektívy manažmentu rizík, čo vlastne bolo cieľom eseje.

Dôležité je, aby sme pri každej činnosti, ktorú vykonávame uvažovali, čo nás môže ohroziť, ako veľmi nás to môže ohroziť a ako sa vyhnúť realizácií tohto ohrozenia. „Ako si nezlomiť ruku ?“, je otázka, ktorú by si mal klásť každý manažér rizík.

Použitá literatúra

1. Addison, T., Vallabh, S.: Controlling software project risks: an empirical study of methods used by experienced project managers. In: *ACM International Conference Proceeding Series*, Port Elizabeth, South Africa, Vol. 30 (2002) 128-140.
2. Armour, P.G.: To plan, two plans. In: *Communications of the ACM*, Vol.48 , No. 9 (2005) 15 – 19
3. Boehm, B.W.: Software Risk Management principles and practices. In: *IEEE Software*, Vol. 8, No. 1 (1991) 426 – 435
4. Cohen, C.F., Birkin, S.J., Garfield, M.J., Webb, H.W : Managing conflict in software testing. In: *Communications of the ACM*, Vol.47 , No. 4 (2004) 68 – 73
5. May, L.J.: Major causes of software project failures., voľne dostupné na internete : <http://www.stsc.hill.af.mil/crosstalk/1998/07/causes.asp>
6. Rettig, M., Simons, G.: A project planning and development process for small teams. In: *Communications of the ACM*, Vol. 36 , No. 10 (1993) 45 - 55
7. Ropponen, J., Lyytinen, K.: Components of software development risk: How to Address them ? A project manager survey. In: *IEEE transactions on software engineering* , Vol. 28, No. 2 (2000) 98 – 112
8. Stoneburner, A., Gougen, A., Feringa, A.: *Risk Mamagemen Guide for Information SystemTechnologiést* . National Institute od Standarts and Technology , U.S Department of Commerce, 2002.

9. Wallace , L., Keil, M. .: Software project risks and their effect on outcomes. In: *Communications of the ACM*, Vol.47 , No. 1 (2004) 76 – 81

Annotation

How not to break your arm

In 1989 Harry W. Boehm identified top ten risks of software development. These risks are still actual. Conditions and the environment of a small project are different. Bounds between the team members are not strong. Their motivation does not depend on a salary. Mitigation of possible risks is crucial for the project success. The essay brings new view to these risks in a small project environment. We will try to identify, which risks are the most dangerous. We compare different methods to identify risks. We focus on persons and risks related with them. We use project risk analyze to improve project control. We discuss the risk impact on the project planning. We will compare classic methods of risk mitigation with optimistic planning methods. We determine the impact of both methods on the project plan.