

NIE JE PROJEKT AKO PROJEKT, NO RIZIKÁ SÚ HROZBOU PRE KAŽDÝ

*Neexistuje len také riziko, že by žiadne riziko
nevzniklo.*

*Zvažuj riziká, oplatí sa to viac ako byť porazený.
(George S. Patton)*

Lenka Baková

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

Abstrakt. Neoddeliteľnou súčasťou každého softvérového projektu je manažment rizík. Mnohokrát sa stretáme s názormi, že analýza a plánovanie rizík nie sú až tak dôležité. Ale proces životného cyklu programu je veľmi zložitý, to prináša so sebou množstvo rizík. Rizikám treba predísť, alebo pri najmenšom ich eliminovať. Musíme ale najprv vedieť, že nám vôbec hrozia, musíme poznať pravdepodobnosť, s akou nastanú a samotné dôsledky rizika. Neskôr môžeme pristúpiť k naplánovaniu ich riešenia. A práve na toto nám slúži manažment rizík. V mojej eseji sa budem snažiť poukázať na najčastejšie riziká v praxi a na najčastejšie riziká v študentskom projekte. Pokúsím sa ich zanalyzovať, uvedomiť si ich dopady a navrhnem ako eliminovať ich výskyt v softvérovom projekte prípadne ako ich celkom vyriešiť. Zamyslím sa nad tým, prečo sa vlastne tieto dve skupiny rizík, študentské a praktické, od seba líšia. V eseji uvediem aj možné metódy pre analýzu a plánovanie rizík a jednu z nich aj podrobnejšie priblížim. V závere zhodnotím potrebu manažmentu rizík a pripomeniem z môjho pohľadu dôležité riziká.

Kľúčové slová: riziko, projekt, tím, študent, prax, analýza rizík, plánovanie rizík, Monte Carlo, Maticová metóda

Úvod

Ludia sa väčšinou snažia robiť veci čo najlepšie. O to viac, ak nie sú platení od hodiny. Preto môžeme počuť názor, že plánovanie rizík je strata času, lebo žiadne nehrozí. Takto optimisticky rozmýšľajú nielen programátori. Myslia si, že ak sa budú snažiť a poctivo pristupovať k práci, nič ich nezaskočí. No opak je pravdou.

Autor zdroja [1] hovorí, že plánovať riziká by sme mali aj pri kopaní jamy, lebo nikdy nevieme, na čo v zemi natrafíme. Teda ak starej mamy sľúbime, že jej vykopeme na záhrade jamu na kompost, v horšom prípade nám hrozí, že bez plánovania rizík ju vykopeme neskôr, ako sme sľúbili, prípadne sa nám ju nepodarí vykopáť vôbec. Stará mama bude iste sklamaná.

No čo nám hrozí pri vytváraní softvérového projektu bez plánovania rizík? Následky neúspešne skončeného projektu nezahladíme zmrzlinovou tortou a kávičkou ako v prípade sklamanej starej mamy. Výsledkom neúspešne skončeného softvérového projektu môže byť peňažná ujma, degradácia v práci alebo strata dôvery zákazníka, čo môže byť v niektorých prípadoch osudové.

Akým rizikám sa vyhýbať? Na ktoré si dať pozor? Možno pred definovaním jednotlivých rizík by som sa zamyslela nad ozdravením pracovného prostredia, nad komunikáciou a motiváciou členov tímu. Vyriešením týchto otázok eliminujeme pravdepodobnosť vzniku viacerých rizík. Jednotlivým rizikám sa budem venovať neskôr.

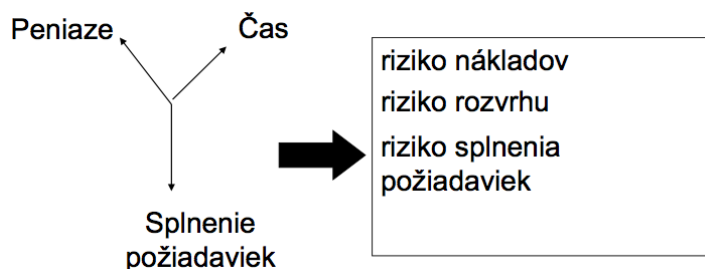
Čo vlastne rozumieme pod pojmom riziko v softvérovom projekte?

Všeobecné vysvetlenie rizika podľa [2]: Riziká sú nežiadúce udalosti, faktory a hrozby, ktoré majú negatívne dôsledky

Ak sa ale spýtate staršieho človeka, možno vám odpovie ako mne moja stará mama: "Riziko je porušovanie zákona."

A teda aký je zákon pri tvorbe softvérového projektu? Je to PSC. Nemyslím tým Poštové Smerovacie Číslo, ale trojicu *Peniaze - Splnenie požiadaviek - Čas*. Tento magický trojuholník (pozri Obr.1) nás upozorňuje na tri základné oblasti rizík:

- prekročenie plánovaného časového limitu na vývoj a dokončenie projektu
- prekročenie finančného rozpočtu pre projekt
- nenaplnenie požiadaviek zákazníka, či už ide o kvalitu alebo o množstvo vlastností výsledného produktu



Obr. 1. PSČ

Manažment rizík

Manažment rizík podľa [8] je nástroj na identifikovanie rizík, ktoré môžu znížiť úroveň komplexnej bezpečnosti. Je prostriedkom na analýzu rizík, ich hodnotenie, klasifikáciu, stanovenie priorít a následne aj ich znižovanie. Stáva sa neoddeliteľnou súčasťou procesu zvyšovania bezpečnosti a má významný podiel na ochrane človeka, materiálnych hodnôt, ale aj kultúrnych a duchovných hodnôt a v neposlednom rade tiež životného prostredia pred účinkami negatívnych dôsledkov krízových javov.

Úlohy manažmentu rizík

Podľa zdroja [2] sú úlohy manažmentu rizík nasledovné:

- Identifikácia rizík
- Hodnotenie rizík
- Analýza rizík
- Spracovanie rizík
- Sledovanie rizík

Riziká v projektoch z praxe

Rizík si môžeme definovať nekonečne vela. Kludne si môžeme vymyslieť riziko, že príde ďalšia doba ľadová a zamrzne nám navždy harddisk a stratíme všetky dáta. Zdravý rozum vie, že pravdepodobnosť tejto situácie je veľmi malá a intuitívne ju ignorujeme. Pán Boehm [3] identifikoval pred rokmi desať najčastejších rizík v softvérových projektoch. Zaujímavé je, že pri enormne rýchlym vývoji informačných technológií dokázal rebríček Boehmových rizík prežiť (pozri Tab.1.).

Tab. 1. Rebríček Boehmových rizík

10 najčastejších rizík v softvérových projektoch podľa Boehma	
1	Nedostatok personálu
2	

3	Vytvorenie inej (než požadovanej; zlej) funkcionality
4	Vytvoreni
5	Pozlátenie systému
6	Neustále (nepretržité) zmeny požiadaviek
7	Nedostatky v externe vytvorených moduloch
8	
9	Nedostatok zdrojov pri samotnom riešení projektu
10	Precenenie technológie

Nedostatok personálu

Je veľmi tenká hranica medzi tým, kedy má projekt nedostatok členov a nestíhajú a tým, kedy má projekt priveľa členov, čo spôsobuje určitým spôsobom chaos a tak isto nestíhajú. Neplatí tu stare známe : “čím viac tým lepšie”. Nie vždy je teda šťastným riešením tohto rizika pridať nového člena tímu, pretože tohto člena treba zaučiť, musí sa zoznámiť so všetkými technológiami a štandardami, čo môže projekt ešte viac zabrzdiť. V takej situácii si vždy treba uvedomiť, koľko času nám zostáva do konca projektu a či nový člen vôbec “stihne” byť produktívny. Nie len ja si myslím, že často schodnejšia cesta pri snahe eliminovať toto riziko, je urobiť všetko preto, aby tých málo členov odvádzalo excelentnú prácu. Treba im vytvoriť zdravé pracovné prostredie, príjemnú atmosféru a hlavne treba ich maximálne motivovať, aby sa do práce tešili a robili ju s radosťou.

Nerealistické rozvrhy a rozpočty

Riziko je spôsobené známou pravdou, že programátori sú veľkí optimisti. Eliminovať riziko je možné tým, že budeme vytvárať plán s chladnou hlavou a budeme sa riadiť podrobnými odhadmi, ktoré sú založené na starších projektoch podobného typu. Pri vyvíjaní treba používať čo najväčšie množstvo hotových “vecí”, to nám výrazne skráti čas vývoja. Hotovými vecami myslím knižnice, algoritmy alebo hotové moduly, ktoré sme vytvárali už skôr pri iných projektoch. Tu sa mi žiada pripomenúť, že je dôležité, aby moduly v softvérovom produkte boli vysoko súdržné a voľne zviazané.

Vytvorenie nepožadovanej funkcionality

Výrazným dôvodom, prečo toto riziko vôbec je, je fakt že viazne komunikácia medzi tímom a zákazníkom. Je potrebné sa s ním pravidelne stretávať a oznámiť mu každú zmenu funkcionality, ktorú sa chystáme robiť. Myslím, že práve komunikáciou sa môže znížiť možnosť výskytu tohto rizika takmer na nulu.

Vytvorenie nevyhovujúceho používateľského rozhrania

Ako v predchádzajúcom prípade, môže nás zachrániť jedine komunikácia. Je veľmi dôležité, ak nie nutné, počas vyvíjania projektu aj “prototypovať” a vytvoriť presné scenáre práce s GUI. Toto treba konzultovať so zákazníkom.

Pozlátenie systému

Kľúčom k riešeniu v tomto riziku je podľa mňa opäť komunikácia. Je dôležité, aby zákazník vedel, aká funkcionálnosť sa implementuje do produktu, aby s tým súhlasil, ba čo viac, aby ju považoval za nutnú. Ak máme pocit, že zákazník váha, treba ho presvedčiť o tom, že daná funkcionálnosť je veľmi dôležitá. Tým pádom, keď si to všetko so zákazníkom vydiskutujeme, nie je možné, aby sme systém pozlátili, z jeho pohľadu nepotrebnými funkciami.

Sústavné zmeny požiadaviek

Keď rozmýšľam nad týmto rizikom a nad jeho riešeniami, zisťujem, že je veľmi ťažké ho eliminovať. Keďže zmeny vychádzajú zo strany používateľa a nevieme odhadnúť, kedy asi prídu. Osobne si myslím, že ten používateľ to nerobí naschvál, jednoducho si neuvedomuje, aké má následky jeho neustále "vymýšľanie", preto by som ja odporúčala hneď na začiatku stanoviť si pravidlá a vysvetliť mu, ako sa veci majú. Prípadne by som sa snažila aj do zmluvy zahrnúť, do kedy je možné meniť požiadavky vzhľadom na termín ukončenia projektu. S týmto možno zákazník súhlasiť nebude, preto by som navrhovala "nahovoriť" ho na to, že ak bude mať potrebu meniť požiadavky aj po onom spomínanom termíne, tak manažér tímu zvolá rokovanie o zmene respektíve posunutí dátumu ukončenia projektu.

Nedostatky v externe vytvorených moduloch

Výskyt rizika eliminujeme, ak externé moduly, knižnice alebo algoritmy pred použitím poriadne analyzujeme, otestujeme ich efektívnosť a zhodnotíme, či by boli pre náš projekt prínosom.

Nedostatky v externe zabezpečovaných úlohách

Nenapadá mi žiadne vlastné riešenie tohto rizika. Dočítala som sa ale, že sa rieši v praxi pomocou auditov a kontroly sa vykonávajú ešte pred pridelením úloh.

Nedostatok výkonu pri riešení projektu v čase

Riziko je spôsobené používaním nedostatočne analyzovaných a testovaných modulov ako aj pozlátením systému. Ak je nám známa požiadavka na výkon, treba ho vyskúšať napríklad pomocou simulácie a prototypovania. Ak požiadavka na výkon nie je, podľa mňa treba "po nej ísť", teda spýtať sa na ňu zákazníka.

Precenenie technológie

V tomto prípade pomáha dôkladné analyzovanie a prototypovanie. Nemali by sme dopustiť, aby v návrhu bola funkcionálnosť, ktorú nie je možné implementovať.

Riziká v študentských projektoch

Riziká vyskytujúce sa v školských projektoch sa mierne líšia od rebríčka rizík, ktoré som uviedla vyššie v kapitole Riziká v projektoch z praxe. Je celkom zaujímavé sledovať tieto odlišnosti a analyzovať ich, zamýšľať sa, prečo vznikajú.

V súčasnosti sa stále viac výskumov sústreďí na študentské projekty ako na celok. Jedna z mnohých štúdií [8] analyzovala riziká z 76 projektov v akademickom roku 2006/07 a 2007/08 (pozri Tab.2.). Posudzované projekty mali 5 až 8 členné riešiteľské tímy zložené zo študentov 3 a 4 ročníkov. Riziká sa analyzovali na základe reportov. Pôvodne bolo v hre 79 projektov, ale dva tímy odovzdali nedostačujúci report a jeden tím report neodovzdal vôbec. Priemerná práca na jednom projekte bola 1217 človekohodín. Vzorka projektov bola celkom rôznorodá. Obsahovala projekty webových aplikácií pracujúcich s databázami, projekty desktopových aplikácií pre Windows a Linux, projekty aplikácií pre mobilné telefóny.

Tab. 2. Rebríček študentských rizík

Riziko	Počet projektov	Percentuálne zastúpenie
Nesprávne plánovanie	47	61%
Nástroje a skúsenosti	46	61%
Technické problémy	40	53%
Práca a štúdium počas projektu	34	45%
Motivácia	27	36%
Choroba a sociálne problémy	26	34%
Komunikácia	24	32%
Požiadavky	24	32%
Problémy súvisiace s klientom	18	24%
Proces	15	20%
Tretia strana	15	20%
Problémy súvisiace s prácou v tíme	14	18%
Ukončovanie	14	18%
Dokumentácia	9	12%

V nasledujúcich podkapitolách sa budem snažiť vyjadriť postupne k rizikám z tabuľky, ktoré majú väčšie ako 30%-né zastúpenie vo vzorke projektov. Pokúsim sa diskutovať o tom, prečo sa líši ich postavenie v rebríčkoch rizík zameraných na študentské softvérové projekty a na softvérové projekty.

Nesprávne plánovanie

Podľa výskumu muselo veľké množstvo tímov upraviť svoj pôvodný plán. Podľa môjho názoru je to tým, že študenti majú málo skúseností s prácou v tíme. Nemajú experta na odhadovanie a plánovanie, preto nedokážu navrhnuť plán približujúci sa realite.

Toto riziko v sebe odzrkadľuje určitým spôsobom aj všetky ostatné riziká, pretože to sú práve tie dôvody, prečo sa nestíha plniť pôvodný plán. Nie je to ale ani zďaleka len záležitosť študentského projektu, tomuto riziku zodpovedá riziko druhé v poradí v tabuľke Tab.1.

Nástroje a skúsenosti

Výskumníci sa domnievajú, a ja s nimi súhlasím, že priemerný študent dokáže produktívne pracovať s nástrojom, ktorý pozná už dlhší čas, napríklad jeden alebo dva mesiace. Navyše pri práci v tímovom projekte sa často stáva, vravím z vlastnej skúsenosti, že v projekte je používaný viac ako jeden nový nástroj. To môže študentom spôsobovať istým spôsobom chaos. Čím viac nových nástrojov súčasne sa študent učí používať, tým dlhšie mu to trvá a tým väčšie komplikácie mu to spôsobuje.

Technické problémy

Toto je veľmi zákerné a nepredvídateľné riziko. Často sa podľa môjho názoru s ním nepočíta a je tendencia ho vôbec nenaplánovať. Ale práve preto, že je také nepredvídateľné je nutné ho naplánovať. Týmto rizikom sa myslia veci ako výpadky internetu, hardverové chyby alebo aj samotné "dožitie" hardvéru, ako napríklad matičných dosiek, displejov, procesorov....

Dopad tohto rizika je rôzny v závislosti od technológií každého tímu. Katastrofálne následky môže mať, napríklad ak tím nepoužíva verziovací systém. Vtedy môže tím prísť o cenné data, čo má v mnohých prípadoch fatálne následky. Drvivá väčšina tímov dnes používa mnoho podporných prostriedkov, ktoré eliminujú práve fatálne následky tohto rizika.

V každom prípade, ja by som odporúčala ako pomerne jednoduché zmiernenie tohto rizika, mať vždy po ruke aspoň jeden záložný stroj s nainštalovanými potrebnými prostriedkami.

Toto riziko v projektoch z praxe nemá zastúpenie, a myslím si, že preto, lebo na klasickom pracovisku je určite minimálne jeden stroj navyše, čím je hrozba rizika zažehnaná.

Práca a štúdium počas projektu

Pravdou je, že mnoho študentov popri škole pracuje. Pracujú v rôznych oblastiach, no skoro pravidlom je, že študenti informatiky a informačných technológií pracujú vo "svojom" IT sektore. To je na jednej strane veľkou výhodou, kvôli skúsenostiam, ktoré tam získajú, no na druhej strane, je to veľmi náročné pre daného študenta. Okrem toho, že sa musí popri práci na projekte sústrediť na iné predmety v škole, musí im venovať čas, porozumieť im, robiť zadania ako klasický študent, musí riešiť aj veci spojené s prácou a teda musí v hlave "držať" veľmi veľa rôznych informácií o termínoch o úlohách a o samotných technológiách, ktoré v rozrobených projektoch používa.

Ako predísť tomuto riziku vieme všetci, treba pracovať ideálne len na jednom projekte, prípadne na čo najmenšom počte projektov. No to však podľa môjho názoru nie je možné a ani nikdy nebude, nakoľko študenti nemôžu ovplyvniť počet projektov, do ktorých "sú zatahnutí", aspoň čo sa týka školských záležitostí.

Toto riziko sa v tabuľke rizík z praxe (pozri Tab.1.) z pochopiteľných dôvodov nenachádza. Ak by sme chceli, môžeme si ho v praxi predstaviť ako situáciu, keď člen tímu pracuje na viacerých projektoch, poprípade vo viacerých tímoch.

Motivácia

Toto riziko je podľa výskumu až na piatom mieste. Ja osobne by som ho odhadovala možno ešte vyššie v rebríčku rizík. Myslím, že motivácia ako taká je jedno z najsilnejších rizík a určitým spôsobom sa odzrkadľuje vo veľkej miere na výsledom softvérovom produkte. Ľudia sú rôzni. U niekoho je ťažké motiváciu získať, u niekoho je ťažké si ju udržať. Ľudia, pre ktorých je ťažké oboje, robia projekt s veľkou nechufou, kazia morálku v tíme a neprispievajú ku kvalite produktu. Existujú ale aj ľudia, pre ktorých je ľahké motiváciu získať a aj si ju udržať. To sú tí ideálni tímoví hráči, ktorí pôsobia ako hnacie motory pre tím. Myslím, že naplánovať riešenie tohto rizika môže byť niekedy poriadne náročné.

Ak vidím, že niekto z tímu stráca motiváciu, treba ihneď zasiahnuť. Každá osobnosť má totiž inú silu a sú ľudia, ktorí dokážu so sebou stiahnuť veľmi veľa slabších, teda ak nezasiahneme hneď a niekto začne šíriť "negatívnu energiu" môže sa stať, že so sebou stiahne aj iných. Ale ako zasiahnuť? Na každého platí niečo iné. Preto je ťažké vymyslieť motivujúce faktory pre ľudí, ktorých nepoznáte.

Aby motivujúci faktor pozdvihol hladinu motivácie u človeka, musí mu byť "ušíty na mieru". Musíte vedieť, čo má konkrétny človek rád a na čom si zakladá. U niekoho je to uznanie, bonusové body, dobrá známka, cena za snahu, u niekoho sú to peniaze... Sú aj ľudia, ktorí potrebujú najprv spadnúť, aby sa mohli postaviť. Ich naštartuje poväčšine len "negatívne ladený motivujúci faktor". Teda musí sa stať niečo zlé, musia dostať "príučku", aby si uvedomili, že sa majú snažiť a aj sa snažiť chcú, lebo po páde sa chcú dostať na pozíciu, v ktorej sa nachádzali.

Následky tohto rizika sú výrazne citeľné. Lebo, ak človek nie je dostatočne motivovaný, prácu nerobí na 100 percent. Práve preto je nesmierne dôležité nepodceniť toto riziko a vypracovať jeho dôkladnú analýzu, lebo takmer s pravdepodobnosťou 1 nastane, čo sa týka aj projektov z praxe.

Choroba a sociálne problémy

Toto riziko nastane tak isto ako riziko nedostatočnej motivácie skoro určite minimálne u jedného z členov tímu. Je veľmi ťažké eliminovať ho. Myslím si ale, že možno ak by sa manažér tímu snažil udržiavať v tíme kľudnú bezstresovú atmosféru nebolo by to na škodu veci. Samozrejme je veľmi vhodné, ak vedenie tímu organizuje členom športové aktivity v rámci firmy, utuží to "zdravého ducha" aj kolektív. Toto však v študentskom projekte možné nie je. Vlastne, ak sa nad tým zamyslím možné by to bolo, no neverím, že sa to v blízkej budúcnosti bude praktizovať.

Komunikácia

Komunikácia okrem toho, že je to jedno z rizík v softvérovom projekte, je jeden z najčastejších kameňov úrazu všade okolo. Eliminácia tohto rizika podľa mňa nie je až taká náročná. Najdôležitejšie je stretávať sa ako kompletný tím často a rozprávať sa, ako o pracovných záležitostiach, tak aj o súkromných, aby sa utužili vzťahy medzi členmi tímu.

Je dôležité povedať svoj názor, lebo ak ho nepovieme, nikto sa ho nedozvie a to môže byť na škodu veci. Netreba nechať iných, nech si domýšľajú. Murphyho zákony fungujú

dokonale a zo skúsenosti viem, že ak očakávam, že si dotyčný niečo domyslí, tak si domyslí pravý opak.

Komunikácia viazne tak v študentských projektoch ako aj v projektoch z praxe, je to dané tým, že študenti aj členovia tímov v praxi sú stále len ľudia.

Požiadavky

Toto riziko ja osobne z vlastných skúseností považujem za najznámejšie učebnicové riziko. Zle definované požiadavky od zákazníka a ich nepochopenie na strane vývojárskeho tímu je "nočná mora" každého softvérového projektu. Pri tomto riziku, tak isto ako aj pri motivácii mám pocit, že by som ho ja osobne zaradila vyššie. Pravdepodobne má túto pozíciu práve preto, že ide o študentský projekt a zadania teda aj požiadavky na výsledný produkt stanovujú vyučujúci. Toto riziko môže mať veľmi nepríjemné následky, ak zistíme neskôr, že sme vlastne robili niečo, čo sme robiť nemali.

Mám nepríjemnú skúsenosť práve s nepochopením požiadaviek od zadávateľa. Šlo o školský "miniprojekt" teda zadanie. Nemala som pocit, že by som zadanie pochopila zle, preto som nemala potrebu ísť sa uistiť, že ho chápem dobre. Pri prezentácii hotového zadania som sa však dozvedela, že som nepochopila istú časť zadania, čo ma dosť prekvapilo. Podľa môjho názoru, ak vývojár nepochopí požiadavky, je to vždy, aspoň čiastočne, chyba oboch strán a za tým si stojím. Práve preto ma veľmi zaskočilo, že sa cvičiaci začal vysmievať z môjho "hlúpeho ženského myslenia", on zrejme nebol toho názoru ako ja, že sú v tom "zamotané" obe strany. Možno krátke poučenie na konci tohto minipríbehu: "Ak ste si aj 100%-ne istý správnym pochopením požiadaviek, vždy sa spýtajte a uistite. Nemôže to byť na škodu veci."

Poučiť sa

Hovorí sa, že len hlúpi ľudia sa učia na vlastných chybách. Stotožňujem sa s týmto názorom, a preto si často pripadáam hlúpa. Stovky štatistík hovoriacich o chybách a rizikách softvérových projektov sú voľne dostupné na internete. Analýza a plánovanie rizík je už značnú dobu téma rôznych výskumov. Prečo je to tak? Podľa mňa práve preto, aby sme sa z toho poučili. Hlavne musíme uveriť, že sa to týka aj nás a nášho projektu.

Informácie o priebehu predošlého projektu sú zahrnuté v záverečnom reporte alebo v záverečnej správe. Ja osobne by som však preferovala iný zdroj informácií ako sú reporty, čím nechcem naznačiť, že sú nepotrebné.

Informácie rôzneho druhu veľmi názorne zobrazia vizualizácie projektu. V dnešnej dobe môžeme vizualizovať prakticky akýkoľvek pohľad na projekt. Za veľký prínos považujem spraviť vizualizáciu evolúcie zmien softvéru. V oblasti manažmentu rizík môže podobná vizualizácia odhaliť to, že zdanlivo malá chybička, ktorú je tendencia zanedbať, ovplyvnila celý rad činností, pričom sa jej na začiatku dalo predísť. Vizualizácia poskytuje tiež rýchly prehľad o tom, pri akých úlohách vzniká najviac chýb a preto sa na ne dá lepšie pripraviť.

Keď si reporty, vizualizácie alebo iné zdroje informácií dôkladne zanalyzujeme a uvedomíme si riziká, ktoré nastali, uvedomíme si, aká je pravdepodobnosť výskytu rizika, čo ho spustilo, ako sa prejavovalo a čo spôsobilo. To sú základné charakteristiky rizika.

Čo ďalej?

Je veľa spôsobov, ako pristupovať k manažmentu rizík. Nasledujúci je najklasickjší a aj najviac zaužívaný.

Maticová metóda

Keď už máme zoznam rizík aj s ich charakteristikou, teda vieme pravdepodobnosť výskytu, poznáme škody, ktoré riziko môže spôsobiť a udalosť, ktorá ho spúšťa, môžeme využiť maticovú metódu [1].

Riziko v nej vieme namapovať na jeden zo štyroch kvadranov matice [pravdepodobnosť X dopad škody]. Pravdepodobnosť je číslo od 0 po 1 a dopad škody je vyjadrený škálou napríklad V - vysoká, S - stredne vysoká a N - nízka. Každému identifikátoru v škále pridáme číslo od 0 po 1, pričom 0 je N a 1 je V. Keď vynásobíme pravdepodobnosť s týmto číslom, dostaneme číslo charakterizujúce závažnosť rizika. Okrem závažnosti máme identifikované spúšťače rizika a máme pripravené aj možné riešenia. Teraz je na manažmente sledovať spúšťače závažných rizík, keď nastane situácia, že sa spúšťač aktivizuje, pristúpime k preddefinovanému riešeniu.

Škála v rámci matice môže byť aj podrobnejšia. Je na konkrétnom manažmente, akú škálu zvolí a ako jednotlivé stupne pomenuje. Takmer vždy sa však používa jednoduchá škála, teda len matica 2 x 2. Podľa zdroja [1] sa táto škála osvedčila v praxi ako najideálnejšia.

Existujú aj mnohé iné prístupy ako manažovať riziká ako napríklad Monte Carlo [5,6] Simulácia, alebo Zem-Mesiach Model [7], ale ako som už spomenula, uvedená metóda je najznámejšia a podľa mňa najlogickejšia a dobre pochopiteľná.

Záver

V eseji som sa snažila objasniť pojmy ako riziko a manažment rizík. Mojim cieľom je, aby si vývojári a manažéri v tímoch uvedomili, aké je to nesmierne dôležité robiť analýzu rizík.

Uviedla som 10 najčastejších rizík, ktoré som sa snažila rozanalyzovať a navrhnúť im riešenia, tak isto som uviedla časté riziká študentských projektov a sledovala som odlišnosti medzi týmito rebríčkami rizík.

Fakt, že skúmaním rizík sa zaoberajú rôzne výskumy svedčí o tom, že riziká tu sú a sú vážnym problémom. Treba sa nimi zaoberať. Ponúkam preto aj podrobnejší náhľad na jednu z metód. Maticová metóda je najviac časom overená a používa sa pri mnohých softvérových projektoch. Som toho názoru, že je obzvlášť vhodná pre rozsahovo menšie projekty. Možno práve pre také, aký je tímový projekt a ako manažérka rizík nášho tímu, som sa rozhodla použiť práve túto metódu pri plánovaní a analýze rizík. Považujem ju za najviac pochopiteľnú a prehľadnú.

Na záver ešte raz spomením citát, ktorý som si vybrala : „Zvažuj riziká, oplatí sa to viac ako byť porazený.“. Nejdem teraz rozoberať, prečo som si ho vybrala a prečo sa mi zdá vhodný. Nechám na čitateľovi, nech si ho vysvetlí sám. Je to podľa mňa najlepšia cesta, ako ho donútiť premýšľať o tejto problematike aj po dočítaní tejto poslednej vety.

Použitá literatúra

1. Staníček, Z., Hajkr, J., Řízení projektu, zavádění IS do organizací, 2005
2. Bielíková, M.: Manažment v softvérovom inžinierstve. 1999
3. Boehm, B. W.: *Software Risk Management: Principles and Practices*, 1991.
4. Ren-hui, L., Feng-yong, Z., *Model Identification of Risk Management System*, IEEE, 2008
5. Croll, J., C., *Cost & Schedule Risk Analysis in Major Engineering Projects*, Eastern Software Publishing
6. Feng, N., Li, M., Gao H., *A Software Project Risk Analysis Model Based on Evidential Reasoning Approach*, IEEE, 2008
7. Wang Y., , Jia J.,, Qu Q., *The "Earth-Moon" model on software project risk management*, IEEE, 2010
8. Athee, T., Pornanen, T., *Risks in Students' Software*, IEEE, 2009

Annotation

Every project is different, but risks threaten all of them

Risk management is an inseparable part of every software project. We often meet with the views that risk analysis and planning are not so important. But the process of the life cycle of a program is very complex and it brings many risks with it. Risks must be avoided, or at least eliminated. But first, we must know that threaten at all, we must know the probability that they will occur and the consequences of the risks. Later we may proceed to plan their solutions. And this is where risk management helps us. In my essay I will try to highlight the most frequent risks in practice and most frequent risks in student project. I will try to analyze them, to realize their implications and to propose how to eliminate their presence in a software project or how to solve them completely. I will think about the reasons, why in fact the two risk groups, student and practical differ. In the essay, I will explain possible methods for risk analysis and planning and I will give more details on one of them. In the end, I will assess the need for risk management and point out risks that are important from my point of view.