

NIEKDE UZNÁVANÝ, INDE ZATRACOVANÝ

Riziko je všadeprítomné. Je len na nás, akým spôsobom sa s týmto faktom dokážeme vyrovnáť.

Ján Trebuľa

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

Abstrakt. Manažment rizík je dôležitá súčasť manažmentu softvérových projektov. Prečo je tak problematické uplatniť teoretické základy riadenia rizík v praxi? Odpoveď by mohli priniesť výsledky empirickej štúdie, ktoré rozoberám v tejto eseji. V eseji taktiež predstavujem a uvažujem nad modelovacím rámcom vývoja softvérových projektov, ktorý je riadený cieľmi. Nakoľko môže byť takýto rámec efektívny? Pre aké tímy je vlastne určený? Úvahy nad podobnými otázkami spolu s potenciálnou budúcou prácou sa nachádzajú v záverečnej časti eseje.

Kľúčové slová: softvérové inžinierstvo, manažment rizík, vývoj softvéru, modelovanie rizík, riadenie cieľmi,

Úvod

Pojem riziko podľa Webster's 1913 Dictionary predstavuje hazard, nebezpečenstvo alebo ohrozenie. Ja riziko vnímam ako skutočnosť, že nie vždy musí všetko vyjsť podľa plánu a v každom okamihu existujú nepredvídateľné faktory, ktoré sa môžu či už negatívne alebo pozitívne prejaviť na výsledku činnosti resp. stanovenom celi. Faktory, ktoré dokážeme ľahko predvídať sa však medzi riziká nezaraďujú.

Oblasť, ktorá sa explicitne zaoberá rozpoznávaním, analyzovaním ale aj riešením rizík sa nazýva manažment rizík. Za jeho úlohu považujem minimalizáciu strát, ktoré môžu byť spôsobené dôsledkom nepredvídateľných faktorov a udalostí. O jeho aplikovanie do oblasti spojenej s tvorbou softvéru sa postaral Barry W. Boehm. Vo svojej práci [1] na prelome osemdesiatych a deväťdesiatych rokov minulého storočia

identifikoval procesy, ktoré by mal správny manažment rizík vykonávať. Napriek tomu, že tieto procesy boli neskôr mnoho krát modifikované, aby čo najlepšie odzrkadľovali a riešili aktuálne potreby pri tvorbe softvéru, ich pretavenie do praxe považujem len za čiastočne uskutočniteľné.

Manažment rizík – prečo áno alebo prečo nie?

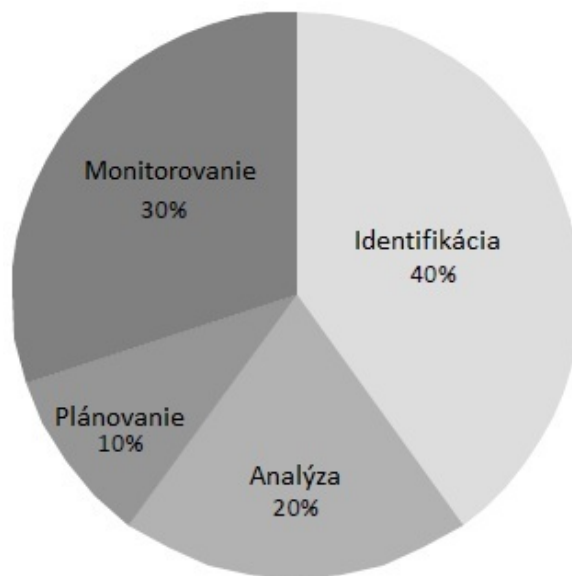
Podľa empirickej štúdie [3] vykonanej pomocou osemnástich skúsených projektových manažérov sa zistilo, že vo všeobecnosti je povedomie o manažmente rizík dobré. Prečo potom štatistiky hovoria, že približne 15,5% projektov končí neúspechom? Dôvody môžu byť rôzne.

Podľa môjho názoru je najväčší problém práve v pretavení teoretických vedomostí do praxe, ktorá je pre rôzne typy projektov viac špecifická. Teoretické vedomosti častokrát opisujú jeden konkrétny príklad podľa stanoveného (ideálneho) scenára. V praxi sa však môžu vyskytovať faktory, od ktorých sa pri teoretickom pojednávaní o manažmente rizík abstrahuje. Medzi tieto faktory zaraďujem napríklad rôzne (i)raciálne požiadavky zo strany zákazníka. Nie raz som sa v praxi stretol so situáciou, kedy zákazník požadoval nelogické resp. nesprávne navrhnuté časti systému a podobne. Samozrejme, že úlohou zberača požiadaviek je okrem iného aj vysvetlenie, odporúčenie a tzv. usmernenie zákazníka, no niekedy napriek tomu trvá na svojom. Môže ísť napríklad o situáciu, kedy zákazník požaduje rozšírenie funkcionality pre už existujúci systém, ktorý bol navrhnutý nesprávne.

Za ďalší z dôvodov považujem fakt, že manažment rizík je často krát považovaný za drahý luxus, ktorý sa v čase krízy vypúšťa z projektu ako jeden z prvých. Samozrejme tento krok sa môže zdať z krátkodobého hľadiska logický, no z dlhodobého prináša so sebou isté riziko. Situácie, v ktorých je prínos manažmentu rizík viditeľný považujem za menej jednoznačné. V situáciách kedy je všetko v poriadku a prebehlo podľa plánu to vyzerá tak, že v podstate žiadny manažment rizík nebol potrebný. Naopak, keď projekt z dôvodu nejakého rizika pohorí, dáva sa to za vinu zlému alebo žiadnemu manažmentu rizík.

Zo štúdie [3] ďalej vyplýva, že problémom je práve rozsah ľudského úsilia vynaloženého pre manažment rizík. Z Boehmova definovaných procesov je možné vidieť, že ich vykonávanie je dosť nákladné a predstavuje monotónnu prácu. Na základe osobných skúseností môžem len súhlasiť, že v praxi sa považuje za dôležitejšie plánovanie a kvalita než riadenie rizík. Práve preto viem potvrdiť, že pre manažment rizík sa často krát vyhradí len minimum dostupných prostriedkov určených pre daný projekt.

Manažéri v prieskume [3] uvádzajú, že najviac úsilia v rámci manažovania rizík je potrebné vynaložiť práve pri ich identifikácii. Naopak za najjednoduchšiu fázu považujú plánovanie (Obr. 1).



Obr. 1. Podiel miery úsilia pri jednotlivých fázach manažmentu rizík.

Podľa mojich doterajších skúseností v tímovom projekte som taktiež zatiaľ najviac úsilia venoval identifikácii možných rizík. Túto fázu považujem za najdôležitejšiu hneď z niekoľkých dôvodov. Jednak na ňu nadväzujú všetky ostatné a preto pri jej zlyhaní môže zlyhať celý proces riadenia rizík. Okrem toho tento proces prirovnávam k hľadaniu niečoho, „čo nevieme ako vyzerá“. Domnievam sa, že výsledný negatívny vplyv môže byť len dôsledkom niečoho hlbšieho, „ukrytého vo vnútri“. Naopak, keď už viem „s čím mám dočinenia“, ľahšie to budem vedieť analyzovať a naplánovať preventívne kroky alebo potenciálne riešenie.

Monitoring resp. dohľad nad samotným procesom v kontexte manažmentu rizík považujem taktiež za nákladný, pretože pre svoje vykonávanie nás núti v každom čase využívať časť z obmedzených prostriedkov (určených napr. pre samotný vývoj). Tým pádom však môže „brzdiť“ celkový proces vývoja.

Analýza a plánovanie v kontexte manažmentu rizík podľa mňa vyžadujú menší podiel úsilia, pretože ide o procesy, ktoré sa vykonávajú pre dané riziko len raz. Ako náhle je už riziko identifikované, jeho analýza a plánovanie prebiehajú z časti implicitne. Myslím si, že v tomto prípade platí pravidlo: „Keď už viem, kde by mohol potenciálny problém nastať, budem si naň dávať väčší pozor“.

Okrem vyššie spomínaných faktorov, ktoré sa podieľajú na problematickom uplatnení manažmentu rizík v reálnych projektoch, existuje prístup, ktorý sa na túto problematiku pozerá z iného uhla pohľadu [2]. Za rizikové časti procesu vývoja softvéru považuje všetky ľudské aktivity. Ľudia môžu robiť chyby, zlé odhady alebo rozhodnutia a pod. Toto všetko pozitívne vplyva na rast potenciálneho rizika. Zástancovia tohto pohľadu tvrdia, že kľúčová je častá komunikácia s koncovými používateľmi vyvíjaného softvéru počas každej etapy jeho vývoja. Nemenej podstatné je vnímať manažment rizík ako celistvý pohľad, ktorý zahŕňa technické aj netechnické aspekty vývoja.

Osobne si myslím, že tento prístup sám o sebe neobsahuje žiadne nové faktory, ktoré negatívne vplyvajú na proces vývoja softvéru. Skôr ide len o celkové zhrnutie resp. zaobalenie tohto problému z hľadiska vplyvov aktivity človeka na daný projekt. Na druhej strane tento pohľad považujem za zaujímavý. Vďaka nemu môžem identifikovať jednotlivé riziká naprieč celým vývojom projektu (riziká, ktoré sa môžu prejaviť pri práci zberača požiadaviek, analytika, návrhára, vývojára atď.). Ďalej dokážem vďaka takémuto pohľadu nájsť potenciálne riešenia vzniknutých problémov pomocou jednoduchej úvahy. Napr. v prípade rizika, že projektový manažér zle odhadne náročnosť projektu, sa mu pokúsím poskytnúť čo najpresnejšie informácie o tomto projekte a pod.

Ďalej nemôžem nesúhlasiť s názorom, že v rámci prevencie je dôležité čo najčastejšie konzultovať s budúcim používateľom systému. Čím menší prírastok v danom projekte bude potrebné po konzultácii prerobiť, tým menšie úsilie bude treba vynaložiť na prácu spojenú s požadovanou úpravou. Je potrebné však dávať pozor, aby veľkosť prírastkov, ktoré chceme konzultovať s budúcim používateľom nebola príliš malá. V takom prípade by to mohlo spôsobiť škody na rôznych úrovniach. Jednak by sme príliš častými konzultáciami mohli demotivovať samotného používateľa resp. zákazníka (dôsledkom by napr. mohlo byť zníženie jeho budúcej ochoty bližšie spolupracovať počas ďalšieho vývoja). Okrem toho ide o problém hospodárenia s prostriedkami vyhradenými pre daný projekt. Čím viac sa budem venovať konzultáciám a iným nevyvojárskym činnostiam, tým menej prostriedkov zostáva pre samotný vývoj.

Manažment rizík riadený stanovenými cieľmi

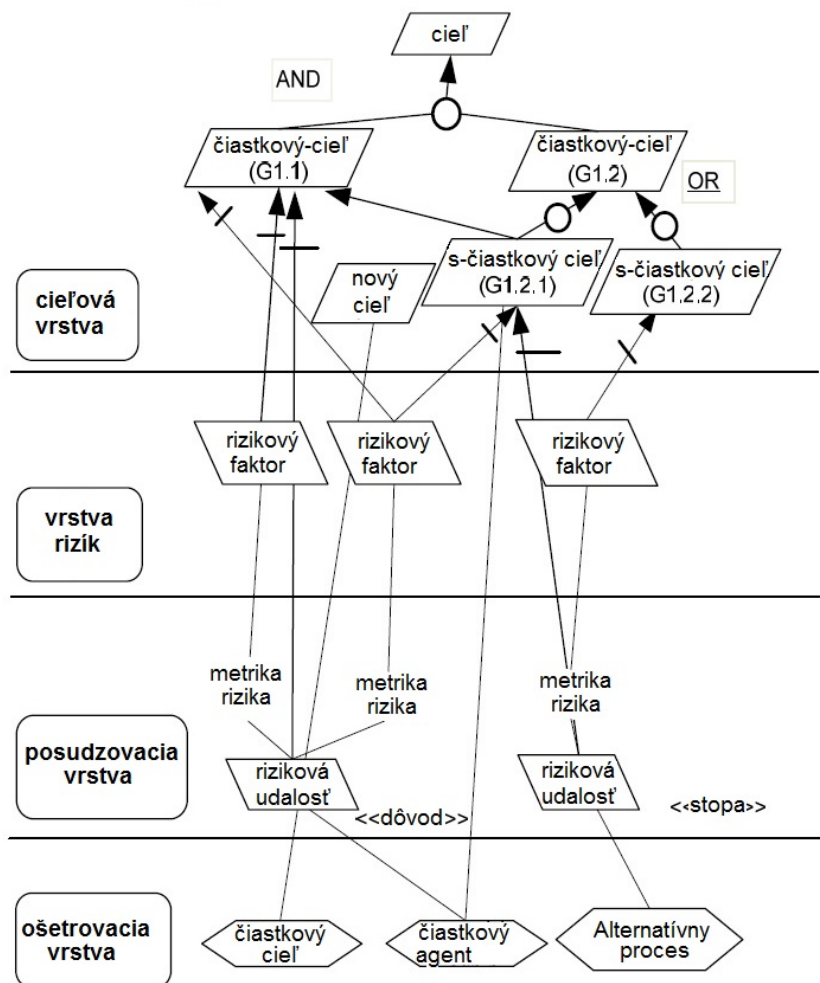
Na princípe posledného spomínaného prístupu k manažmentu rizík vznikol aj modelovací rámec vývoja softvérových projektov riadený cieľmi (angl. Goal-Driven Software Development Risk Management Model - GSRM) [2]. Jeho úlohou je podpora manažmentu rizík najmä vo fáze získavania požiadaviek a skoršej implementácie. Cieľom je detekcia rizík v čo najskoršej fáze projektu za účelom minimalizácie ich dopadu na samotný projekt. Aby nedochádzalo ku konfliktom viacerých uplatnených prístupov, rámec je tvorený štyrmi vrstvami.

Vrstva cieľov (angl. Goal layer) slúži na identifikáciu, spracovanie a modelovanie cieľov. Je založená na vývoji komponentov z perspektívy úspechu projektu. Existuje niekoľko smerov ako definovať úspech projektu napr. splnenie dohodnutých obchodných cieľov, dokončenie projektu na čas a v rámci dohodnutého rozpočtu, realistické požiadavky, realistické odhady času a nákladov a pod. V tejto vrstve sa identifikujú ciele na základe vyššie spomínaných kritérií. Takéto ciele môžu mať vzájomne rôznu úroveň abstrakcie. V prípade potreby je možné cieľ s vyššou úrovňou abstrakcie rozdeliť na niekoľko cieľov nižšej úrovne. Úspešné splnenie nižších cieľov určite naplní aj pôvodný abstraktný cieľ.

Vrstva rizík (angl. Risk-obstacle layer) má za úlohu identifikovať potenciálne riziká už počas skorého vývoja softvéru. Identifikované riziká sú vnímané ako prekážky, ktoré bránia naplneniu cieľov. Jedna takáto prekážka môže negatívne vplyvať na viacero cieľov. Identifikácia týchto prekážok prebieha pomocou existujúcich zoznamov, dotazníkov a brainstorming-u so všetkými zainteresovanými stranami. Dôraz sa kladie práve na dotazníky založené na prvých vyvinutých komponentoch.

Posudzovacia vrstva (angl. Assessment layer) slúži na presné označenie identifikovaných prekážok. Hlavným cieľom je analyzovanie rizikových udalostí, ktoré môžu byť spúšťané identifikovanými rizikovými faktormi. Každá riziková udalosť je charakterizovaná dvoma parametrami: pravdepodobnosť, že nastane a jej závažnosť. Závažnosť kvantifikuje jej negatívny dopad. Jeden rizikový faktor môže zodpovedať viac ako jednej rizikovej udalosti rovnako ako jedna riziková udalosť môže negatívne vplyvať na viacero cieľov. Pomocou takejto reprezentácie je možné modelovať rôzne situácie konkrétnych kombinácií spomínaných vplyvov. Vrstva používa metriky rizík na určenie pravdepodobnosti rizikových udalostí meraním rizikových faktorov. Posudzovacia vrstva nakoniec zoraďuje riziká na základe ich pravdepodobnosti a závažnosti ich dopadu a negatívnym vplyvom na požadované ciele.

Ošetrovacia vrstva (angl. treatment layer) identifikuje možné kontrolné opatrenia a vyberá z nich najvhodnejšie z hľadiska zmiernenia rizík a dosiahnutia cieľa. Ako náhle predošlé vrstvy identifikujú ciele, rizikové faktory a rizikové udalosti potom táto vrstva hľadá optimálne protiopatrenia na elimináciu resp. zníženie dopadu z hľadiska času a nákladov. Kontrolné opatrenia sú reprezentované rôznymi agentmi napr. ľuďmi alebo nástrojmi tak, aby mohli efektívne napomáhať k plneniu cieľov. Pred zavedením zvolenej kontrolnej akcie je však potrebné urobiť analýzu nákladov a výnosov, aby navrhované riešenie nemalo negatívnejší dopad než scenár bez neho. V tomto rámci na to slúži prepojenie štvrtej a druhej vrstvy. Na Obr. 2 je možné vidieť model celého rámca GSRM.



Obr. 2. Model rámca vývoja softvérových projektov riadeného cieľmi [2].

Vyššie opisovaný rámec považujem za efektívne riešenie manažmentu rizík najmä pri rozsiahlych projektoch, ktoré zahŕňajú veľké množstvo účastníkov. V prípade malých projektov si myslím, že použitie tak komplexného prístupu by bolo kontraproduktívne. Okrem toho si myslím, že realizácia tak zložitého procesu je sama o sebe dosť nákladná a je potrebné dávať pozor, či jej uplatnenie bude celkovo prínosné.

Napriek tomu si myslím, že by bolo zaujímavé vybrať z neho niektoré princípy a skúmať ich efektívnosť v menších projektoch (napr. školský tímový projekt). Osobne by som skúsil aplikovať členenie manažmentu rizík na nezávislé vrstvy ako v spomínanom rámci. Pri dôkladnom dokumentovaní jednotlivých vrstiev vidím prínos v tom, že viacero rizík sa v softvérových projektoch opakuje. V prípade už identifikovaných rizikových faktorov resp. rizikových udalostí by sme z predchádzajúcich skúseností mali model, ako sa situácia môže vyvíjať a nemuseli by sme ho tvoriť odznovu. Stačilo by ho len adaptovať do konkrétneho projektu nastavením potrebných parametrov (cena, rozsah atď.) a pod.

Nemuselo by ísť len o modely rizikových faktorov a udalostí ale aj o modely samotných cieľov projektu a ich scenárov pre jednotlivé riziká. Takáto množina

modelových situácií by sa získavala a naplňala postupne počas riešenia prvých projektov. V počiatkových fázach by sa naplňanie takejto množiny mohlo javiť drahé a neefektívne, no postupom času by sa pomer nových a už zaznamenaných scenárov obrátil. Vtedy by sme pri identifikácii už namodelovaného cieľa, rizikového faktora či rizikovej udalosti mohli použiť existujúci model, čo by mohlo aspoň čiastočne ušetriť prostriedky.

Záver

Manažment rizík slúži na elimináciu resp. minimalizáciu nežiaducich vplyvov pri procese vývoja softvérového projektu. Jeho realizácia v reálnych projektoch nie je triviálna a častokrát si vyžaduje veľké množstvo prostriedkov vyhradených pre daný projekt. Aj to je jeden z možných dôvodov jeho častého zlyhania. Zlepšenie úspešnosti projektov je možné realizovať napr. uplatnením modelovacieho rámca GSRM, ktorý je založený na riadení cieľmi. Jeho uplatnenie je však efektívne najmä pri rozsiahlych projektoch. Ďalší postup práce vidím práve v modifikovaní a prispôbení tohto rámca pre menšie projekty.

Použitá literatúra

1. Boehm, Barry W. Software Risk Management: Principles and Practices. IEEE Software. Jan 1991.
2. Islam, S. Software Development Risk Management Model – A Goal Driven Approach, 2009, pp. 5–8.
3. Odzaly, E.E., Des Greer, P.S.: Software risk management barriers: An empirical study. Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, 2009, pp. 418 – 421.

Annotation

Somewhere respected, elsewhere deprecated

Risk management is an important part of software project management. Why is so difficult to implement theoretical principles of risk management in practice? Empirical study results, which I analyze in this essay could respond. Moreover, in this essay I present and consider modeling framework of development software project that is directed by objectives. Whereas can be this framework effective? For which teams is actually aimed? Reflections about similar issues along with potential future work are situated in final section of this essay.