

ZABOČILI SME NA ZLEJ KRIŽOVATKE, ČO ĎALEJ?

Radšej skôr, ako neskôr a lepšie neskôr, než nikdy.

Peter Bako

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

Abstrakt. *Plánovanie softvérového projektu prináša určité riziká, ktoré sa niekedy nedajú dopredu určiť. Odhady trvania prác môžu byť problematické najmä pre menej skúsených manažérov, napr. aj tých, ktorí vedú školské projekty ako Tvorba softvérového systému v tíme. Preto sa môže stať (a často sa aj stáva), že projekt nenapreduje požadovaným smerom, tempom, nastávajú časové sklzy. Vtedy je úlohou manažérov, resp. vedúceho projektu rozhodnúť sa pre alternatívy, ktorými môžeme dohnať stratu, vrátiť sa na správnu cestu. V eseji sa zamýšľam nad krízovými situáciami v projektoch a analyzujem alternatívy, ktoré sa odporúčajú pri podobných krízových situáciách (prechod na extrémne programovanie, atď.). K analýze týchto riešení pristupujem z hľadiska predmetu Tvorba softvérového systému v tíme.*

Kľúčové slová: *Plánovanie, riziká, extrémne programovanie*

Úvod

Len málo ľudí si vie predstaviť život bez plánov. Ľudia prišli na to, že plány motivujú, prispievajú k efektívnosti práce, umožňujú sumarizovať dosiahnuté ciele a zavádzajú určitý „poriadok“ do činností, projektov. Preto si plánujeme nasledujúci deň, týždeň, víkendy, školu, voľný čas ale aj niekoľko rokov dopredu. Niektorí ľudia si dokonca plánujú aj dôchodok.

Aj plánovanie softvérového projektu je vo väčšine projektov neodmysliteľné. Je naozaj dôležité vedieť, kedy sa čo má spraviť, dokedy máme stihnúť aké aktivity a ako ďaleko sme pokročili v projekte, vzhľadom na plán. Aj v tomto prípade plány predstavujú

akúsi kľúčovú úlohu, ktorá prispieva k väčšej efektívnosti, motivácii a dostatočnej kontrole nad projektom. K menšine, kedy sa plánovať neoplatí, patria spravidla len projekty malého rozsahu, na ktorých pracuje málo ľudí, väčšinou len jeden človek. Ani v týchto prípadoch to nejde bez plánov, avšak s tým rozdielom, že ľudia si ich udržiavajú v hlavách a vedia odhadnúť, že projekt sa neoplatí, plánovať, pretože je jednoduchý. Projekt na predmet Tvorba softvérového systému v tíme (ďalej TP) patrí do tej skupiny projektov, kedy je plánovanie takmer nevyhnutné, preto alternatívu bez plánovania nebudeme brať do úvahy.

Čo je vlastne ten plán?

V úvode sme si načrtli, že život človeka je vo všetkých oblastiach spätý s plánovaním. Ani softvérové projekty si nevieme predstaviť bez plánov. Podľa [5] projektový plán slúži pre projektový tím, aby informoval vývojárov o tom, čo má byť kedy hotové. V [2] sa uvádza, že plán slúži aj na stanovenie vzťahu so zákazníkom. Ak ideme ešte ďalej, zistíme, že plán slúži vlastne na komunikáciu medzi zákazníkom a vývojármi. Zákazník si povie, čo chce mať spravené, plánovač to spracuje do plánu a vývojár sa podľa toho riadi.

Plánovanie softvérového projektu má (alebo by aspoň mal mať) zväčša na starosti jeden človek (v prípade menších projektov) – ten, čo má s plánovaním najväčšie skúsenosti. Úloha je pre neho o to ťažšia, čím má menej skúseností s podobnými projektmi. Náročnosť plánovania narastá taktiež s rozsahom a zložitou softvérového riešenia. Do hry môže vstúpiť aj nedostatočná znalosť schopností členov tímu, čo vedie k zlým časovým odhadom. V tom lepšom prípade sa na úlohy prideli viac času, ako je potrebné, vtedy dochádza k plytvaniu času. V horšom prípade sa projekt nestíha dokončiť v rámci časového harmonogramu (v dôsledku zlých odhadov, prípadne aj iných faktorov). Vtedy to so sebou prináša určité sankcie. Firmy musia zvýšiť svoje náklady, zákazník môže udeliť pokutu z omeškania, v prípade školských projektov to znamená zníženie hodnotenia až nepridelenie zápočtu.

Z uvedeného vyplýva, že hoci plánovanie nepredstavuje značnú časť z celkovej práce na projekte, napriek tomu môže prinášať veľké riziko.

Je to riskantné. Skutočne?

Riziko je prítomné v každom softvérovom projekte. Niekedy vo väčšej miere, inokedy v menšej, ale do hry vstupuje vždy. Môžeme ho charakterizovať ako hrozbu akejkoľvek neželanej udalosti, spojenej s prácou na projekte [4]. Riziko sa dá zhrnúť do dvoch komponentov: očakávaný dôsledok a pravdepodobnosť, že udalosť nastane. Každé riziko sa skladá z týchto dvoch komponentov, ktoré spolu úzko súvisia, aj keď sú absolútne odlišné [4].

Riziko je nežiaducim faktorom pri softvérových (ale aj hociktorých iných) projektoch. Niekedy sa dá dopredu predvídať a prispôbiť sa mu, inokedy nás prekvapí úplne nečakane. Aby tých prekvapení bolo čo najmenej, je vhodné pri plánovaní sa zaoberať analýzou rizík [4]. Táto činnosť si vyžaduje nemalé úsilie, čo sa môže zdať zbytočné pre manažerov alebo vedúcich projektov, avšak výhody plynúce z primeranej analýzy rizík dosvedčujú jej vhodnosť. Medzi tie výhody patrí [4]:

- **Väčšia istota, že sa projekt dokončí** – manažment rizík počíta s prípadmi, ktoré môžu ohroziť dokončenie projektu a poskytuje alternatívy pre zvládnutie situácie
- **Nižšie náklady a menej chaosu** – adekvátna analýza rizík dáva dojem väčšej kontroly nad problémami. Minimalizuje sa množstvo práce strávenej prepracovaním a tým znižujú náklady.
- **Podpora manažmentu a pridelenie väčšej priority** – projekt, ktorý má veľmi neistú budúcnosť, bude mať menšiu prioritu ako projekt, ktorý má jasné ciele a je pripravený aj na nečakané situácie. Medzi súbežnými projektmi má väčšiu šancu, že dostane požadované zdroje a práve aj týmto sa znižuje riziko.
- **Ďalšie výhody pri manažmente** – ideálne je, ak sú vo firme ideálne obsiahnuté projekty vyššou aj nižšou mierou rizika. Pri organizácii projektov sú údaje o rizikách kľúčové.
- **Lepšie plánovanie** – pomocou analýzy rizík sa odhaľujú slabé a neisté stránky projektu, čomu sa dá prispôbiť plánovanie.
- **Vymedzenie rezerv** – analýza rizík nám odhaľuje neistoty a tým nás núti vymedzovať si rezervy. Projekt vždy vyžaduje určité časovo ohraničené úsilie. Pri plánovaní sa predpokladá určitý horizont dokončenia, avšak pri započítaní rizík sa tento horizont predlžuje. To znamená, že ak plánujeme projekt dokončiť za 12 mesiacov, ale posledný termín nasadenia je o 14 mesiacov, projekt dokončený v tomto horizonte sa pokladá za úspešný.
- **Podpora riadenia a komunikácie** – tento aspekt úzko súvisí s plánovaním, keďže spoľahlivý plán zahŕňa aj manažment rizík. Členovia tímu sú tým pádom lepšie oboznámení o možných hrozbách a ich následkoch, čo im umožňuje efektívnejšiu a uvedomelejšiu prácu.

Výhody začlenenia vhodnej analýzy rizík sú zjavné a nie je ich málo, napriek tomu je diskutabilné, či je vhodné sa ňou zaoberať aj na projektoch TP. Študenti majú tú nevýhodu, že s podobnou činnosťou zväčša nemajú žiadne skúsenosti. Je otázne, či by táto práca priniesla požadovaný efekt. Na druhej strane môžu získať nové skúsenosti aj z tejto oblasti.

Plán nevychádza

Predstavme si situáciu: Pracujeme na menšom softvérovom projekte. Všetko bolo primerane naplánované, každým dňom sa približujeme k cieľu – dokončeniu projektu. Myslíme si, že je všetko v poriadku, no zrazu vysvitne, že to, čo sme naimplementovali, nie je skutočne to, čo zákazník požaduje.

Predstavme si inú situáciu: Pracujeme na menšom projekte, všetko bolo primerane naplánované, ale zisťujeme, že čím ďalej, tým viac sa vyskytujú komplikácie, čo vedú stále k väčším časovým sklzom. Príde za nami manažér a oznámi, že musíme zvýšiť pracovné nasadenie.

Môže nastať aj iná nepríjemná situácia: práca na projekte pomaly finišuje, ale pri nasadení sa zistí, že miera poruchovosti je veľmi vysoká

Všetky tieto scenáre majú niečo spoločné. Tou je krízová situácia, kedy treba niečo zmeniť. Na ťahu je manažment. Niekedy postačí zmena plánu, inokedy je potrebné zmeniť

prístup k riešeniu problému, manažment, pridať ľudské zdroje, alebo dokonca ísť znovu od začiatku. Tak ako sa máme rozhodnúť?

Čo sa dá robiť?

Na túto otázku nájdeme odpovede okrem iného aj v príručkách a učebniciach softvérového inžinierstva. Naším cieľom je vlastne eliminovať straty spojené s krízovou situáciou, ktoré väčšinou znamenajú nedokončenie projektu načas. Hoci opísané situácie sú rôznorodé, všeobecne sa odporúča niekoľko alternatív:

1. Pridanie zdrojov
2. Ubratie plánovanej funkcionality
3. Personálne zmeny v manažmente
4. Implementácia pohotovostného plánu, preplánovanie

Každý projekt je jedinečný, jeho výsledkom je unikátny produkt. Preto neexistujú dva rovnaké projekty, aj keď riešia rovnaký problém. Prečo? Za projekt považujeme časovo ohraničené úsilie, ktorého výsledkom je jedinečný produkt. Aj keď je zadanie rovnaké v dvoch projektoch, už nie sú rovnaké, pretože v každom sú ľudia s rôznymi skúsenosťami, schopnosťami, zručnosťami. A práve týmto sa vylúči, že výsledky budú rovnaké a naraz. Týmto chcem povedať, že v praxi sa nestretáme s dvomi rovnakými krízovými situáciami, každá má svoje špecifiká, na každú je otázka ideálneho riešenia iná, preto existujú len dohady, čo, kedy a ako riešiť. Ja sa napriek tomu pokúsím ohodnotiť navrhované stratégie a priradiť k nim modelové prípady.

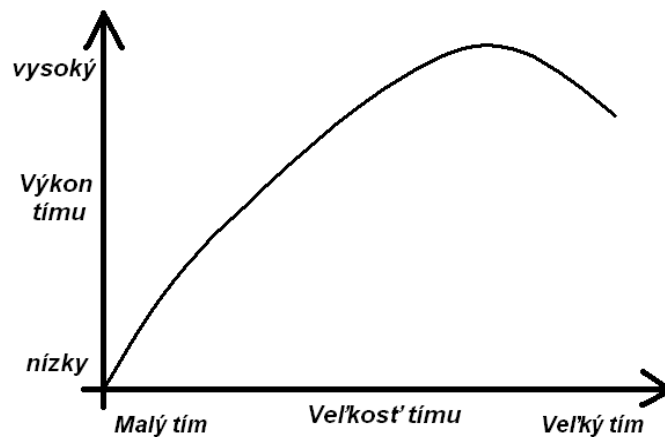
Pridanie zdrojov – menej je viac?

Znie to logicky – nestíha sa, tak si na to zoberme viac ľudí. Veď viac ľudí viac spraví. Nie? Takéto rozhodnutie nemusí byť vždy najšťastnejšie. Samozrejme záleží od veľkosti tímu, resp. rozloženia úloh. Avšak existuje istá hranica, kedy je tím najvýkonnejší. Viac členov nemusí jednoznačne znamenať väčší výkon. Pri veľkých tímoch totiž prichádzajú problémy s organizáciou. Preto je lepšie mať viac menších tímov ako jeden veľký. Podľa [2] najlepšie výsledky vykazujú tímy do 8 členov. Relatívna závislosť výkonu tímu od počtu členov vyjadruje Obr. 1.

Začlenenie ďalších vývojárov do tímu môže priniesť aj iné riziko. Noví členovia totiž nemusia byť zbehlí v danom projekte, ich oboznámenie sa môže viesť k ďalším časovým sklzom. Toto tvrdenie sa zhoduje aj s myšlienkou Brooksa [3]: *Pridanie inžinierov do oneskoreného projektu ho oneskorí ešte viac.*

V prípade projektov na TP táto alternatíva väčšinou neprichádza do úvahy, keďže vedúci projektu nedisponuje voľnými zdrojmi. Avšak podľa môjho názoru tímy na TP zápasia väčšinou s časovými sklzmi, keďže ako to u študentov býva zvykom, všetko robia na poslednú chvíľu. Vtedy už nie je čas na vysvetľovanie a zasväcovanie do projektu. Zabrало by to príliš veľa času.

Táto alternatíva sa môže javiť ako vhodná vtedy, keď sa treba vrátiť veľký krok späť. Vtedy sa počiatočná investícia do zaškolenia nových členov vráti počas dlhšieho obdobia, kedy odvádzajú svoju pridanú hodnotu.



Obr. 1. Výkon tímu/počet členov [2].

Ubratie funkcionality – riešenie v nečase?

Možnosť vynechať niektoré funkcie projektového riešenia nemôže vždy prísť do úvahy. V každom prípade to znamená stratu a aspoň čiastočný neúspech. Táto alternatíva sa využíva v krajných prípadoch, kedy už niet iného východiska. Vtedy sa zákazník dohodne s klientom o ďalšom postupe, väčšinou sa chýbajúce zložky doimplementujú dodatočne.

Z pohľadu TP to je jasné. Znamená to zníženie celkového hodnotenia. Tento prípad ja ani nepovažujem za „riešenie“, skôr len znášanie dôsledkov nedostatočnej práce nesprávneho manažmentu alebo privysokých požiadaviek.

Personálne zmeny v manažmente – ten nový to dotiahne!

Keď projekt nenapreduje správnym smerom/tempom, väčšinou, ako prví si to odnesú manažéri projektu. Niekedy sa stáva, že vedúci sa rozhodne vymeniť niektorých ľudí z manažmentu, keďže vidí nádej na zlepšenie, zvýšenie efektívnosti práce, lepšie riadenie a kontrolu nad projektom, väčšiu motiváciu. Takýto krok so sebou prináša ďalší rad rizík, situácia sa môže aj zhoršiť.

Ťažko povedať, či je vhodné využiť túto alternatívu pri projektoch na TP. Tímy si vyberajú za manažéra člena, ktorý má najväčšie skúsenosti s manažmentom. Výmena manažéra asi nebude postačujúca. Je to samozrejme individuálne pri každom projekte.

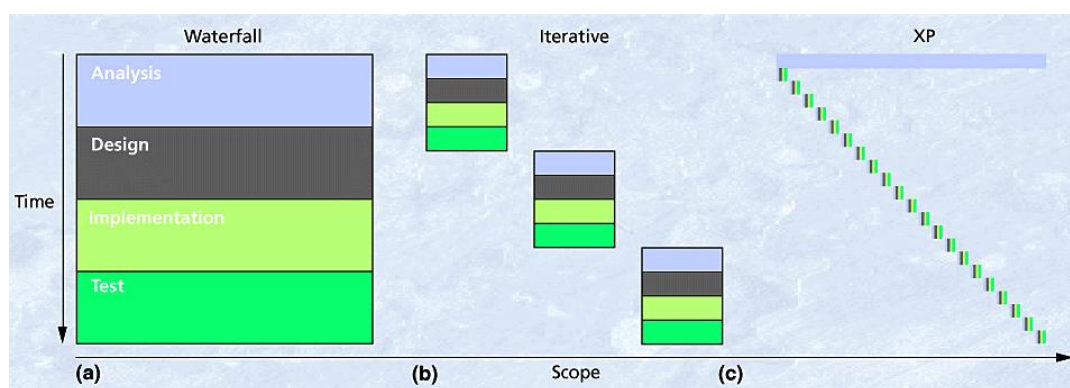
Implementácia pohotovostného plánu, preplánovanie – vyriešime to takto!

Ak bola urobená dôkladná analýza rizík, je možné, že sa s danou krízovou situáciou počítalo. Vtedy sa zväčša prejde na núdzový plán, ktorý zohľadňuje nové skutočnosti. Ak je situácia absolútne nečakaná, znamená to nemalú výzvu pre celý tím. Je potrebné vyjsť z nových predpokladov, identifikovať nové riziká, navrhnúť nové plány ako reakciu na potenciálne hrozby, vytvoriť nové rozvrhy činností [4].

Táto alternatíva sa zväčša využíva, ak sa počas projektu zmenia požiadavky, naplní očakávaná hrozba, alebo sa zistí, že doterajšia filozofia riešenia nebude úspešná.

A čo tak ísť do eXtrémov?

Extrémne programovanie (ďalej XP) je jedna z metód agilného vývoja softvéru, ktorá je určená pre projekty s meniacimi sa požiadavkami, alebo cieľom, ktorý dopredu nie je úplne ujasnený. XP predstavuje alternatívu k manažmentu rizík, pretože sa nesnaží riziká identifikovať, analyzovať a vypracovávať plány v prípade ich naplnenia, ale pristupuje k nim úplne iným spôsobom. Pri tejto metodike sa totiž s rizikom počíta počas celého vývoja projektu takým spôsobom, že jeho metódy sú navrhnuté tak, aby sa riziko úplne potlačilo. Na rozdiel od vodopádového modelu, kde sa každá fáza vývoja (analýza požiadaviek, návrh, implementácia, testovanie) vykonáva po poradí postupne pre celý projekt a kompletne, XP vykonáva tieto aktivity po kúskoch, v krátkych cykloch (Obr. 2) tzv. „stories“ (príbeh).



Obr. 2. Porovnanie modelov vývoja softvéru [5].

Príbehy sa určujú pri analýze, ale aj priebežne, sú silne viazané na požiadavky zákazníka, každý príbeh znamená určitú funkčnosť, musí sa dať implementovať a otestovať. XP sa orientuje na požiadavky zákazníka. Zákazník si určuje, ktoré príbehy sú najdôležitejšie a tie sa potom idú riešiť. Takto môže byť systém nasadený aj s čiastočnou funkčnosťou v krátkej dobe. Každý implementovaný príbeh je otestovaný (aj zákazníkom), pričom systém musí byť v každej fáze vývoja funkčný. Filozofiu XP nám lepšie priblížia hlavné praktiky XP [1]:

- *Plánovanie* – Zákazník sa dohodne na časovom rozvrhu v spolupráci s programátormi. Nové verzie sú vydávané v krátkych intervaloch (rádovo v dňoch až týždňoch)
- *Jednoduchá architektúra* – Systém má spĺňať všetky požiadavky a pritom byť čo najjednoduchší, podľa filozofie „Say everything once and only once.“
- *Testovanie* – testuje sa aj niekoľko krát denne, pričom v každom okamihu musia byť všetky testy úspešné.
- *Programovanie v pároch* – vždy dvojica programátorov pracuje spoločne
- *Priebežná integrácia* – nová funkčnosť je takmer okamžite integrovaná do systému, pričom pri (aj po) integrácii musia byť všetky testy úspešné
- *Kolektívne vlastníctvo kódu* – každý programátor má v každom okamihu prístup ku každému kódu, hoci má možnosť hocičo vylepšiť.

- *Zákazník súčasťou vývojového tímu* – tým sa zabezpečí, že sa skutočne rieši to, čo treba
- *Týždeň má 40 hodín, nie viac* – nadčasy znižujú výkonnosť tímu a signalizujú nejaký problém, ktorý treba riešiť.

Vďaka tejto filozofii sa sľubuje zníženie nákladov, skrátenie vývoja systému, minimalizovanie rizík, ľahšie prispôsobenie zmenám, zlepšenie produktivity a aj iné výhody. Vďaka týmto výhodám sa XP podľa môjho názoru javí ako vhodné núdzové riešenie v rôznych krízových situáciách. Ja som na predchádzajúcich stranách načrtnol 3 problémy:

1. **Nepochopenie požiadaviek** – XP je na to pripravené. Keďže zákazník úzko spolupracuje s tímom, táto hrozba je v podstate vylúčená.
2. **Meškanie harmonogramu** – XP využíva krátke vývojové cykly s postupným uvoľňovaním nových verzií, pričom vždy sa implementuje to, čo je najviac potrebné pre zákazníka. Vďaka úzkej spolupráci zákazníka s tímom sa minimalizuje hrozba, že projekt „zabočí na zlej križovatke“.
3. **Vysoká miera poruchovosti** – vďaka dôkladnému testovaniu programátormi aj zákazníkom sa minimalizuje riziko poruchovosti.

Myslím, že je užitočné zamyslieť sa nad touto alternatívou ešte pri začatí projektu a nielen, keď je už oheň na streche. Môžeme ním predísť nemalým problémom.

Záver

Plánovanie je u reálnych projektov nevyhnutné, dôležité je, ako sa k nemu postavíme. Nie je jedno, aký spôsob vývoja softvéru zvolíme, každý prináša určité riziká. Ak sa hrozby naplnia, veľa záleží na tom, ako sme na ne pripravení. Práve v tom nám môže pomôcť analýza rizík. Napriek tomu sa môže stať, že nás zasiahne čosi nečakané. Popri klasických alternatívach som ako riešenie navrhol extrémne programovanie nielen ako riešenie problémov, ale aj ako prevenciu a alternatívu k analýze rizík.

Použitá literatúra

1. Beck, K.: *Embracing Change with Extreme Programming*. IEEE. 1999.
2. Bieliková, M.: *Softvérové inžinierstvo: Princípy a manažment*. STU Bratislava 2000. ISBN 80-227-1322-8
3. Brooks, F.P.: *The mythical man-month*. Anniversary Edition. Addison-Wesley. 1995
4. Kendrick, T.: *Identifying and Managing Project Risk*, New York, Amacom 2003. ISBN 0-8144-0761-7
5. Rettig, M., Simons, G.: *A project planning and development process for small teams*. Communications of the ACM, Vol.36, No.10, 1993

Annotation

We have turned off on wrong crossroad, what can we do?

Planning software projects brings some risks, which are often hard to uncover. Project time estimation can be difficult especially for managers with lack of experience, e.g. managers of projects on subject „Team project“. Hence, it can happen (and also it happens), that the project does not shape the right way, time lapses can happen. It is the task of the managers to choose the right alternative, which can turn the project on right way and recover the dropout. In the essay, I contemplate the problems of crisis situations in projects and analyze alternatives that are recommended to use on similar situations (extreme programming, etc.). I consider the view point of the subject “Team project”.