Ako vytvoriť „dobrý plán“

ĽUBOŠ PAZDERA

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
Lubos.pazdera@gmail.com

Abstrakt. Esej sa snaží identifikovať najčastejšie chyby, ktoré tvorcovia plánov v softvérových projektoch robia. Dokument nielen poukazuje na rizikové faktory vplývajúce na softvérový projekt, ale aj ponúka riešenie ako možno vytvoriť „dobrý plán“. Esej pojednáva o spôsobe vytvárania realistického i pesimistického plánu a hodnotí oba tieto prístupy. Pri tvorbe realistického plánu je rozoberaný konkrétny model plánovania etáp a ľudských zdrojov. V časti popisujúcej pesimistický plán je rozoberané riešenie, ktoré umožní pokryť prípadné zaostávanie za časovým harmonogramom. V nasledujúcej časti esej kladie dôraz na monitorovanie reálneho stavu projektu a jeho porovnávanie s projektovým plánom, čo vedie k prípadnej aktualizácii plánu. Záverečná časť sumarizuje výsledky práce a zároveň poukazuje na možnosti ďalšieho postupu.

Úvod

Podľa [2] je vytváranie projektového plánu predvídanie budúcich udalostí s cieľom minimalizovať riziká pri behu projektu.

Na priebeh projektu však vplýva množstvo rušivých faktorov. Snáď najmenej ovládateľným faktorom zo všetkých je ľudský faktor. Len veľmi ťažko sa v počiatočných fázach projektu odhaduje, koľko ľudí z realizačného tímu ochorie, či odíde na dovolenku počas trvania projektu. Ďalším faktorom sú prirodzené vlastnosti softvéru, medzi ktoré patria zložitosť a neviditeľnosť. Zo zložitosti vyplýva komplikovanosť komunikácie so zákazníkom a často krát aj skreslenie medzi predstavami zákazníka a tímu vytvárajúceho riešenie. Neviditeľnosť zasa vplýva na nepresnosť odhadov trvania a náročnosti.

Ak však na projekt vplýva také množstvo nepredvídateľných faktorov, dá sa projektový plán urobiť? A ak aj projektový plán urobíme, nakoľko bude vierohodný? Existuje dokonalý, perfektný plán?

Manažment v softvérovom inžinierstve, október 2006, s. 1-9.

Plánovanie, najslabšie ohnivko reťaze

Ak sa dokonalý plán urobiť dá, je hudbou budúcnosti. My si budeme musieť vystačiť s „dobrým plánom“. Pokúsme sa v nasledujúcich riadkoch popísať, čo by v takomto pláne malo byť zahrnuté a čomu by sme sa mali radšej vyhnúť.

Plánovanie má dopad na všetky ostatné fázy procesu tvorby softvéru. Zle vytvorený projektový plán je preto najčastejším dôvodom zlyhania procesu tvorby softvéru. Steve McConnell v [3] popisuje týchto 9 smrteľných hriechov plánovania projektu, ktorých sa treba vystríhať:

1. Neplánovanie – Množstvo problémov v softvérových projektoch je zapríčinené práve nevytváraním žiadnych plánov.
2. Nedostatočné plánovanie – Niektoré projektové plány sú vytvárané s predpokladmi, že nikto z projektového tímu neochorie. Plány ktoré obsahujú zidealizované predpoklady môžu byť pohromou.
3. Nepripravenosť na riziká – Neuvažovanie o rizikách v softvérových plánoch môže mať pre plán fatálne následky.
4. Používanie rovnakého plánu na každý projektu – Každý projekt má svoje špecifiká. Použitie rovnakého plánu na každý projekt znamená ignorovanie špecifík projektu.
5. Nekritické používanie všeobecne uznávaných techník – Navzdory tomu, že unifikované procesy alebo agilné metódy sú odporúčané postupy, nie je možné ich používať bezhlavo na všetky druhy riešení.
6. Dovolenie plánu vzdialiť sa od reality – Akokoľvek dobre vytvorený plán sa môže vplyvom problémov odtrhnúť od reality. Nechať takýto plán neupravený, ho robí nefunkčným.
7. Plánovanie príliš mnohých detailov príliš skoro – Úroveň abstrakcie musí byť v každej fáze zachovaná na príslušnej miere. Snaha naplánovať v počiatočnej fáze projektu systém do úplných detailov je takmer rovnako zlá, ako neplánovať vôbec.
8. Dobehnutie strateného – častá chyba pri zaostávaní za časovým plánom je predpoklad, že sa stratený čas získame v ďalšej fáze.
9. Nepoučenie sa z minulých chýb

Ak chceme vytvoriť „dobrý plán“, musíme reálne odhadnúť riziká projektu, potrebné zdroje, časový harmonogram a počítať s nepredvídateľným. Cieľom plánovania je mimo iného aj vytváranie manažmentu rizík. Uvedomenie si rizík a kalkulácia s nimi nám umožňuje klásť dôraz na kritické aspekty projektu.

Podľa mojich skúseností sú najhorší projektoví manažéri optimisti. Optimistický projektový plán zvyčajne predpokladá, že práce na projekte pôjdu „ako po masle“.

Termíny etáp sú tvorené s heslom „to je jednoduché“ alebo „niečo také sme už predsa robili“. Nie je ťažké si predstaviť akú katastrofu vyvolá sebe menšia komplikácia v procese vývoja.

Poďme sa pokúsiť vytvoriť dobrý plán. Začnime vytvorením realistického plánu, v ktorom triezvo zvážime všetky aspekty a zahrnieme doň manažment rizík. Následne doň vnesme kontrolovaný závan pesimizmu. Verím že práve takto vzniká „dobrý plán“.

Venujme sa jednotlivým etapám osobitne.

Model na vytváranie realistických plánov

V [4] je predostieraný spôsob vytvárania plánov založený na organizácii a plánovaní malých tímov. Postup práce pri vytváraní plánu je nasledovný.

V začiatočných fázach projektu sa vytvárajú projektové tímy a stanovujú sa interné tímové smernice, ktoré dodržiavajú všetci členovia tímov. Minimálny počet ľudí v tíme je tri, maximálny počet je sedem.

Obsadzované role v tíme sú:

- Programátor – zapisuje navrhnutý kód a tak implementuje riešenie
- Doménový špecialista – znalec aplikačnej domény
- Technický recenzent – programátor (iný než človek s rolou Programátor) poskytuje kritiku návrhu a implementácie.

Programátor zapisuje navrhnutý kód a tak implementuje riešenie. Doménový špecialista je znalec aplikačnej domény. Technicky recenzent posudzuje vytvorený návrh a implementáciu. Každá osoba v tíme má pridelenú len jednu z úloh. V prípade väčších tímov môžu byť role obsadené dvoma až troma členmi tímu.

Projekt je v počiatočnej fáze analyzovaný za účelom rozdeliť ho na etapy. Etapa trvá od štyroch do šiestich mesiacov.

Etapa má jasne zadefinovaný začiatok, koniec a ciele, ktoré majú byť počas trvania etapy dosiahnuté. Na konci etapy je vytvorená časť zintegrovaná s doteraz vytvoreným riešením. Za výhodu tohto postupu považujem uchopiteľnosť doteraz vytvoreného riešenia. Fakt, že máme novú verziu ako celok a používame ju ako východiskovú verziu pri ďalších prácach nám dáva istotu, že vytvorené komponenty ako celok naozaj fungujú. Ak máme aktuálnu verziu inkrementálne už počas procesu vývoja, odpadajú obavy z dodatočných problémov s integráciou častí riešenia do funkčného celku.

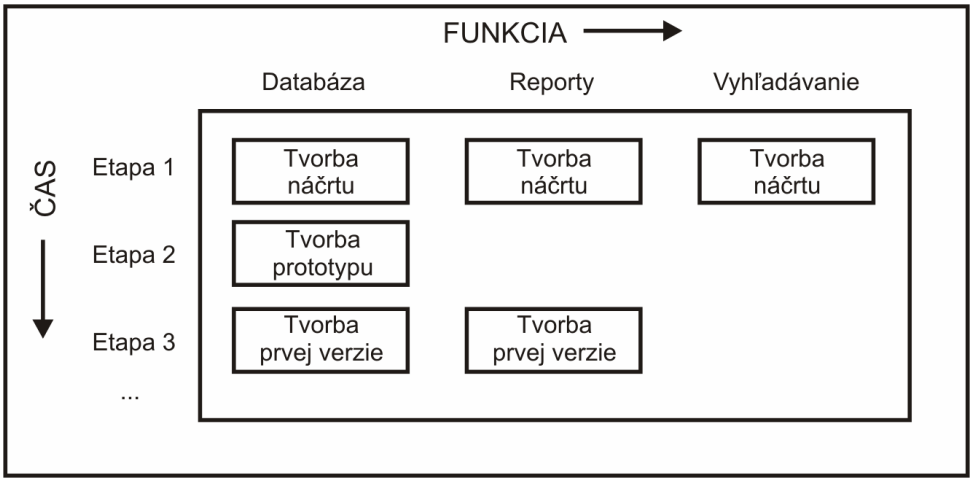
Každá etapa sa delí do fáz:

- Tvorba náčrtu
- Tvorba prototypu
- Tvorba prvej verzie
- Údržba na požiadanie

Cieľom tvorby náčrtu je dodať koncovému používateľovi čo možno najrýchlejšie predstavu, ako bude koncové riešenie vyzerať a tak sa ubezpečiť, že naša predstava o vytváranom riešení korešponduje s predstavou používateľa. Podľa môjho názoru je práve prezentácia náčrtu alebo funkčného prototypu najbezpečnejšou formou overenia správnosti formálnej špecifikácie.

Do náčrtu a ďalej prototypu sa zapracujú pripomienky používateľa až do vytvorenia prvej oficiálnej verzie. Táto je predaná používateľovi a upravovaná na jeho žiadosť.

Obrázok **Obr. 1** zobrazuje deľbu projektu po časovej osi na etapy a po funkčnej osi na komponenty.

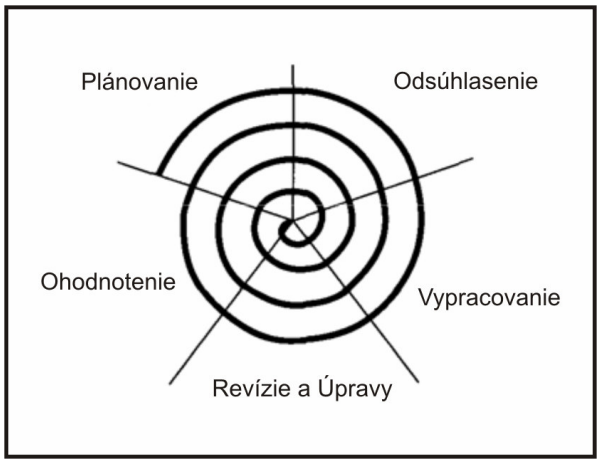


Obr. 1. Ilustrácia deľby projektu po funkčnej a časovej stránke

Po následnom analyzovaní je etapa rozdelená na menšie časti – moduly, ktoré sú prideľované tímom. Rozsah modulu predpokladá približne dva týždne práce pre jedného človeka. Na rozsahu modulu je realizovaný mikromanažment projektu.

Výhody manažovania projektu v takých krátkych časových rozsahoch vidím v rôznych aspektoch. Jednak je jednoduchšie odhadnúť predpokladanú dĺžku práce jedného človeka na dva týždne, ako skupiny ľudí na šesť mesiacov. A tiež sa prípadné meškanie projektu prejaví veľmi rýchlo, čo nám dáva možnosť podniknúť také kroky, aby meškaním nebola ovplyvnená zvyšná časť projektu.

Realizácia modulu má päť míľnikov, ktoré zodpovedajú špirálovému modelu životného cyklu. Opis špirálového modelu vývoja je zobrazený na obrázku **Obr. 2**.



Obr. 2.Špirálový model životného cyklu

Míľnik *plánovanie* je dosiahnutý potom, čo je vytvorený plán na implementovanie a následné testovanie modulu.

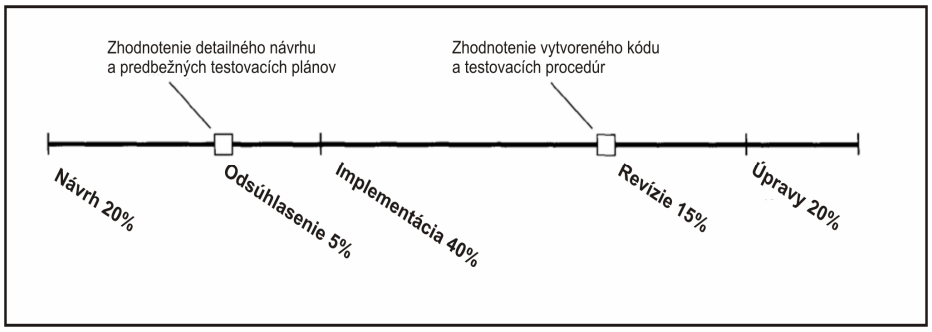
Míľnik *odsúhlasenie* je dosiahnutý po odsúhlasení plánu celým tímom.

Míľnik *vypracovanie* je dosiahnutý v okamihu, keď programátor ukončí implementáciu modulu.

Míľnik *revízie a úpravy* je dosiahnutý úspešným otestovaním modulu.

Míľnik *ohodnotenie* je dosiahnutý po ohodnotení práce na module tím manažérom.

Úprava tohto životného cyklu aj s popisom trvania jednotlivých fáz je zobrazená na obrázku **Obr. 3**.



Obr. 3. Popis fáz pri vytváraní modulov

Míľnik *plánovanie* je pomenovaný výstižnejším pojmom návrh a míľnik *vypracovanie* je pomenovaný implementácia. Míľnik *revízie a úpravy* je rozdelený na dve etapy tak, aby bolo možné vidieť, aká je časová náročnosť každej jednej z nich.

Model na vytváranie pesimistických plánov

Aj najlepšie realistické plány môžu zlyhať. Zakladať proces tvorby softvéru na realistických plánoch znamená veriť, že všetky okolnosti a faktory ktoré vplývajú na projekt vieme správne odhadnúť už v začiatočnej fáze projektu. Takýto prístup je však veľmi odvážny a pre projekt môže mať nemilé následky.

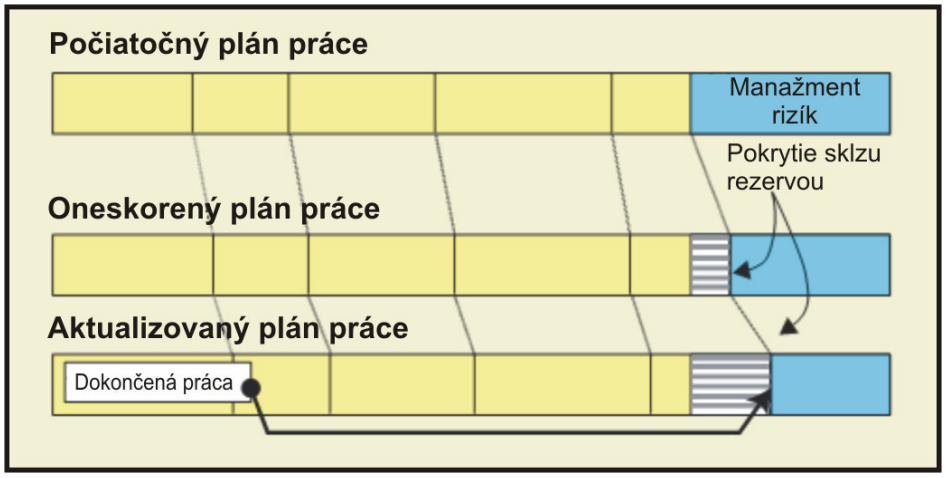
Ak chceme vytvoriť robustný plán, ktorý bude vedieť pohltiť istú mieru rizika spojeného s nepredvídateľným, musíme predpokladať, že postup práce pôjde horšie než dokážeme odhadnúť. Z tohto dôvodu sa k odhadu trvania vývoja projektu postavíme pesimisticky.

Jeden z modelov pre pesimistické plánovanie môžeme nájsť v [1]. Základom tohto modelu je vytvorenie dvoch plánov s dvoma termínmi.

1. Plán prace
2. Plán prezentovaný zákazníkovi

Plán práce je realistický plán, podľa ktorého postupuje vývoj projektu. Plán prezentovaný zákazníkovi je realistický plán + kryté oneskorenie projektu. Oneskorenie je čas, ktorý si vyžiadajú nepredvídateľné vplyvy na projekt. Môže byť vypočítané buď ako konštantné percento z realistického plánu alebo môže zahŕňať známe rizikové faktory z analýzy rizík.

Kryté oneskorenie ktoré sme k realistickému plánu pridali, vytvorí priestor do ktorého môžu prípadné prieťahy vývoja rásť. Model vytvorenia dvoch plánov je zobrazený na obrázku **Obr. 4**.



Obr. 4. Model manažmentu rizík prostredníctvom vytvorenia dvoch plánov

Tri horizontálne pásy popisujú modifikácie plánu projektu. Žlté pole je odhadnutý reálny plán projektu. Modré pole je kryté oneskorenie plánu. Šrafované pole je oneskorenie projektu v jednotlivých fázach. Pre práce na projekte je vždy aktuálny žltý plán. Pre zákazníka je aktuálnym plánom celá dĺžka stĺpca.

Potenciálnym problémom takéhoto modelu je, že ani zákazník ani tím pracujúci na projekte sa nesmie dozvedieť o tom, že existujú dva plány. Zákazník by požadoval dodať softvér k skoršiemu dátumu, lebo by predpokladal, že sme schopný prácu stihnúť skôr než bola dohoda. Tím by si prácu prispôsobil uvoľnenejšiemu plánu a tak by sa risk manažment obsiahnutý v tejto metóde stratil.

Čo ak sa o pláne prezentovanom zákazníkovi dozvie projektový tím? Nebude sa cítiť podvedený? Zrejme áno, ale vždy je tu možnosť vysvetliť tímu pohnútky, ktoré nás k tomuto kroku viedli.

Mať vývoj pod kontrolou

Neexistuje dokonalý plán. Dokonca ani pesimistický plán, ktorý zohľadňuje riziko nepredvídaných prieťahov nie je zárukou ukončenia projektu načas.

Ak sa chceme vyvarovať 6. hriechu, nesmieme dopustiť, aby sa náš plán odtrhol od reality. Ak ustanovíme kontrolný cyklus a únosnú mieru vychýlenia projektu od plánu, môžeme v pravej chvíli plán aktualizovať.

Autori v [4] navrhujú postup, ako kontrolovať plnenie plánov na úrovni modulov. Pri overovaní je využívaný hárok zobrazený na obrázku **Obr. 5**.

Programátor: Názov modulu: Číslo modulu: Odhadovaný čas na dokončenie modulu: Upravený odhad:			
Míľniky	Pôvodný plán	Upravený plán	Aktuálny dátum
1. Začiatok			
2. Návrh pripravený na revíziu			
3. Úpravy návrhu hotové			
4. Kód pripravený na revíziu			
5. Úpravy kódu hotové			
6. Úspešne integrované			

Obr. 5.Hárok používaný pri kontrole stavu prác v moduloch projektu

Na počiatku vytvárania modulu programátor vyplní pole *Odhadovaný čas na dokončenie modulu* predpokladaným časom potrebným na dokončenie modulu a pole *Začiatok* v stĺpci *Pôvodný plán*. Hodnoty vo zvyšných poliach stĺpca sa dopočítajú podľa predpokladaných trvaní fáz. Keďže tieto odhady boli vytvárané bez predchádzajúceho štúdia modulu, môže sa stať, že časový odhad nekorešponduje zložitosti implementácie. Na tento účel slúži stĺpec U*pravený plán***.** Programátor ho

vypĺňa po naštudovaní modulu. V prípade že je treba pozmeniť plány, sú tieto zmeny konzultované s tím manažérom. Po schválení zmien programátor vyplní stĺpec *Upravený plán* novými hodnotami.

Steve McConnell v [3] poukazuje na fakt, že ani dlhoročné skúsenosti s vedením projektu nie sú dôvodom na stratu obozretnosti pri kontrole reálneho stavu projektu: „Ak budem nútený vybrať si medzi expertom v projektovom plánovaní, ktorý nezvažuje pozorne svoj plán a amatérom ktorý dôkladne zhodnotil potreby projektu, vždy si vsadím na amatéra."

Záver

Identifikovali sme najčastejšie problémy, ktorých sa projekt manažéri dopúšťajú. Poukázali sme na problémy, s ktorými sa projekt počas jeho behu stretáva a ktoré sa negatívne odrážajú na plnení stanoveného projektového plánu. Ponúkli sme spôsob ako vytvoriť dobrý plán. Začali sme modelom na vytváranie realistických plánov, kde sme sa sústredili na plánovanie malých tímov. Popísali sme spôsob ako pomocou metodického prístupu vytvoriť realistický plán. Následne sme ponúkli možnosť, ako spraviť realistický plán do istej miery robustný voči nepredvídateľným faktorom.

Popísané riešenie je len jedným z mnohých alternatív ako zvládnuť dynamiku procesu tvorby softvéru. Každý projekt je do istej miery jedinečný a preto sa treba rozhodovať pre použitie tej či onej metódy vždy nanovo. Pre isté projekty je vhodná silne systematická a komplexná metóda ako je RUP, ale pre iné projekty môže byť vhodná práve agilná metóda. Pri niektorých projekty je potrebné minimalizovať náklady, napríklad kvôli ponuke v konkurze a tak nie je možné aplikovať postup pesimistického plánu. Sú však projekty, kedy poistenie sa proti istej miere rizika je na mieste.

Proces tvorby plánu je komplexný proces, v ktorom treba robiť kompromisy. Ak by sme vždy presne vedeli ušiť projektový plán na mieru projektu, vytvorili by sme „perfektný plán", v ktorom kompromisy robiť netreba. Verme, že i tento čas raz príde.

Použitá literatúra

1. Armour, G.P.: The business of software: To plan, two plans. *Communications of the ACM,* Vol. 48, No. 9 (September 2005) 15-19.
2. Krašna, M., Rozman, I., Stiglic, B.: How to improve the quality of software engineering project management. *ACM SIGSOFT Software Engineering Notes,* Vol. 23, No. 3 (Máj 1998) 120-125.
3. McConnell, S.: A The Nine Deadly Sins of Project Planning. *IEEE Software,* Vol. 18, No. 5 (September-Október 2001) 5-7.
4. Rettig, M., Simons, G.: A project planning and development process for small teams. *Communications of the ACM,* Vol. 36, No. 10 (Október 1993) 45-55.

Annotation

How to make a good plan

This paper describes common mistakes made by project managers in process of making project plans. Considers risk factors which influence software project. Proposes solutions instructing how to make a “good plan”. Analyses ways of making realistic and pessimistic plans, discusses particular advantages and disadvantages. Specific model of project and human resources planning is proposed in discussion of realistic plan making. Solution for covering project delay is discussed in part of pessimistic project plan making. It lays emphasis on the project status monitoring and comparing with the project plan, which can lead to project plan actualization. Achieved results are evaluated and extension possibilities are present.