

Je každý dobrý plán naozaj taký dobrý?

PETER CÍCH

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

Abstrakt. Na to, aby mohol byť softvérový projekt úspešný je určite potrebné mať dobre zostavený plán. Existuje široké množstvo najrôznejších metód na vytváranie plánov, ale pri použití ktorejkoľvek z nich môžu vzniknúť problémy, ktoré nás donútia naše plány zmeniť. Žiaden problém nevzniká len tak, samovoľne, práve naopak... Jedným z dôvodov sú chyby, ktorých sa pri plánovaní dopúšťame. A práve týmito chybám sa budem v nasledujúcej eseji venovať. Spoločne sa pozrieme na 9 základných chýb plánovania v softvérovom projekte, na ich príčiny a dôsledky. Ďalej sa pokúsim zakomponovať do tvorby plánov aj ľudský rozmer – človeka, ako neoddeliteľnú súčasť plánovania a vývoja softvérového projektu. Človeka, na ktorom stojí a padá celé plánovanie, od ktorého závisí všetko, pretože ani človek, podobne ako žiaden plán, nie je dokonalý.

Mám svoj plán...

Bolo to pred niekoľkými rokmi, čo som v rádiu zachytil kratučký, ale za to výstižný úryvok jednej pesničky: „Mám svoj plán, na ňom zakladám, som sám sebe pán...“

Je dobré ak má každý svoj *plán*. A je ešte lepšie ak je tento plán aj dobrý. Všetku našu vynaloženú energiu a úsilie je výhodnejšie investovať do našich aktivít na základe nejakého kvalitného plánu, určitej predstavy, ako nechať veci okolo nás len tak nekontrolovateľne a nezáväzne plynúť. Podobne je tomu aj v projektoch, obzvlášť v tých softvérových.

Podobne ako si každý v živote robí plány, či chce alebo nie, je viac-menej nutné, aby plánovanie prebiehalo aj v softvérovom projekte. Plán tvorí totiž základ všetkého. A dobrý plán znamená nepochybne pevnejší základ, a teda aj predpoklad vyššej kvality výsledného produktu. Je potrebné *na ňom zakladať*...

Otvorenou otázkou teda zostáva: Ako vytvoriť dobrý, kvalitný plán? Opísať tvorbu dobrého plánu nie je vôbec jednoduché. To, čo sa môže javiť pri jednom projekte ako dobré a vhodné, nemusí vyhovovať špecifickým podmienkam iného projektu, a naopak. Z tohto dôvodu by som sa preto radšej zameral na iné spoločné črty

plánov – na to, čo v plánoch nie je dobré – na chyby, ktorých sa pri plánovaní dopúšťame.

Pri tvorbe dobrých plánov nesmieme zabúdať ani na najdôležitejší článok, na samotnú „výkonnú jednotku“ – človeka. Všetky veľké veci, a teda aj plány v softvérových projektoch, sa skladajú z menších častí – z plánov každého člena tímu. Tieto plány je potrebné spolu skĺbiť a zosynchronizovať. Je preto veľmi dôležité, aby sa vedel „naplávať“ aj sám jednotlivec. Ak totiž zlyhá, môže so sebou „stiahnuť“ celý tím. Jednotlivci môžu spolu vytvárať tím, avšak neustále zostáva platný fakt, že každý z nich je *sám sebe pán*...

Takto nie!

Pri plánovaní v softvérovom projekte sa určite vyskytnú chyby – nič, a ani nikto nie je predsa dokonalý. Ak však o týchto chybách a nedostatkoch budeme vedieť ešte skôr ako sa vyskytnú, ak ich budeme poznať, dokážeme sa na ne pripraviť, čeliť im, prípadne sa im včas vyhnúť. Budeme schopný im včas predchádzať, a zabrániť tak nechceným komplikáciám a vynútenému – neplánovanému zdržaniu.

Preto by som sa teraz chcel zamerať na niekoľko najzávažnejších chýb plánovania softvérového projektu [4], na ktoré by sme mali pri tvorbe plánu myslieť. Pri tvorbe plánu by sme nemali:

Neplánovať vôbec

Najbežnejšou a najčastejšou chybou pri tvorbe plánov a plánovaní je paradoxne neplánovanie. Absencia plánu spôsobí absolútne nekontrolovateľný vývoj v softvérovom projekte, pretože projekt nebude mať presne určený cieľ a už vôbec nebude *jasná cesta*, po ktorej by sa mal vývoj uberať.

Aby sme mali každý svoj projekt pod kontrolou, a aby sme si boli vedomí, čo, ako a hlavne kedy to chceme dosiahnuť – musíme si vyvíjať plány. Hlavné dôvody prečo vytvárať plány teda sú [2]:

1. redukovať neurčitost' (výsledku projektu)
2. zlepšiť výkonnosť
3. zabezpečiť lepšie porozumenie cieľom projektu
4. vytvoriť základňu pre sledovanie a riadenie práce

Ak budeme mať pre projekt vytvorený plán, bude sa určite uberať iným smerom – lepším, ako keby sme plán nemali.

Nedostatočne plánovať

Neplánovať je zlé. Avšak plánovať len nedostatočne a povrchné nie je taktiež veľmi správne.

Aby boli naše plány dostatočne dobré, musíme plánovať precíznejšie a brať ohľad na viacero aspektov. Plán musí v sebe zahŕňať aj neočakávané situácie. Dobrý plán musí predvídať dopredu. Na dlhej ceste vývojom softvéru nás čaká nespočetné množstvo okolností a nástrah, ktoré môžu ovplyvniť jeho nerušený priebeh.

Podľa môjho názoru sa v každom rozsiahlejšom projekte vyskytne minimálne jeden bod, v ktorom nastane vážnejší problém. Ak však už dopredu predpokladáme výskyt takejto komplikácie, môžeme sa jej vyhnúť. Veľkej časti nejasností sa môžeme vyhnúť lepším a dostatočnejším plánovaním. Preto je potrebné plánovať, ale nielen povrchné. Treba *plánovať dostatočne*.

Nebrat' do úvahy riziká

Riziko je neoddeliteľnou súčasťou každého vývoja softvéru. Nemusí nevyhnutne znamenať niečo zlé a katastrofálne, niečo čo spôsobí predčasné ukončenie projektu. Riziko je len určitý úsek v realizácii softvérového projektu, na ktorý si treba dať väčší pozor a ktorému je nevyhnutné venovať viac času ako ostatným – nerizikovým fázam. Inak nám môže „poriadne skomplikovať život.“

Pri tvorbe dobrého plánu si musíme uvedomiť všetky možné riziká, ktoré by nás mohli zasiahnuť, ak by sme na ne neboli pripravení. Musíme ich vziať do úvahy a následne zakomponovať do nášho plánu, t.j. vymedziť pre ne primeranú časovú rezervu, aby sme mali dostatok priestoru a prostriedkov na vyrovnanie sa s nimi.

Až potom budeme mať plán, ktorý bude „*myslieť na všetko*“, dostatočne dopredu, nevynímajúc ani nepriaznivé, na prvý pohľad možno fatálne, rizikové okolnosti.

Používať rovnaký plán pri každom projekte

Každý jeden softvérový projekt je špecifický a má svoje vlastné črty – je originál. Je preto doslova nezmyslom používať na plánovanie jedného projektu plán z nejakého iného, pravdepodobne predošlého projektu. Každý projekt sa nachádza vo svojom vlastnom prostredí, v danom čase a na danom mieste, a je preto takmer nemožné, aby boli okolité podmienky rovnaké pri dvoch, prípadne viacerých projektoch.

Projektový plán by mal byť „*šitý na mieru*“ pre daný projekt, a nie prevzatý z iného projektu. Ak bol plán použitý pri predošlom projekte, určite už nebude taký dobrý aj pre náš projekt.

Myslím si, že plány podobných softvérových projektov môžu byť do určitej miery podobné a dokonca aj využiteľné, avšak podobnosť neznamená zhodnosť a rovnosť. Treba dôkladne zvážiť, ktorá časť plánu je, a ktorá nie je vhodná pre náš nový projekt. Nemôžeme len tak bezhlavo a bez rozmýšľania používať plány z minulých projektov.

Nekriticky aplikovať všeobecne používané plány

To, že máme k dispozícii aj existujúce, niekým iným vopred pripravené plánovacie scenáre neznamená, že ich musíme nekompromisne použiť aj vo svojom softvérovom projekte. Aj keď sú tieto postupy odporúčané a vo veľkej miere určite využívané v iných projektoch, pre náš konkrétny projekt, v daných podmienkach, môžu byť absolútne nepoužiteľné a nevhodné.

Je preto na mieste zamyslieť sa a dobre si zvážiť, či je rozumné a výhodné použiť v projekte všeobecne osvedčené a uznávané plány, alebo sa nimi len inšpirovať a vytvoriť si svoj „*vlastný dokonalý plán*.“ Len ľudia „zvnútra“ dokážu plne porozumieť procesom prebiehajúcim v danom projekte, a preto nie je ani možné, aby niekto „zvonka“ (niekto tretí) dokázal všetko dokonale pochopiť a naplánovať. Ak niekto nepozná všetky detaily, nie je správne a dobré, aby sme sa bez váhania riadili jeho plánom. Aj keď je možno práve ten jeho plán často aplikovaný a uznávaný, pre náš konkrétny projekt môže znamenať katastrofu. Alebo prinajmenšom neželané komplikácie.

Umožniť plánu divergovať od reality projektu

Divergovať znamená rozchádzať sa, resp. odchyľovať sa od niečoho. Ak náš plán diverguje od reality znamená to, že sa od nej vzdáva a prestáva byť s ňou spojený. Načo je nám potom taký plán, ktorý nemá takmer nič spoločné s realitou, a teda ani s naším softvérovým projektom?

Veď predsa plán je vytváraný pre projekt, a nie opačne – projekt pre plán. Plán je to, čoho sa projekt pridrižiava a podľa čoho sa riadi. Avšak v prípade ak bude plán nereálny a nebude mať s projektom nič spoločné, nebude už mať pre projekt zmysel. Akoby ani žiaden plán nebol...

Ak zistíme, že plán, ktorý používame už nekorešponduje s reálnym stavom projektu je na mieste ho upraviť. Na to, aby mohol byť náš plán dobrý, musí sa neustále meniť a *prispôsobovať realite*.

Veľmi skoro plánovať do veľkých detailov

„Čím viac sa snažíme plánovať do budúcnosti, tým je menšia pravdepodobnosť, že náš plán bude mať aspoň nejakú reálnu hodnotu.“ [3]

Ako som už niekoľkokrát spomenul – plánovať nepochybne treba. Na druhej strane – nie je plánovanie ako plánovanie. *Všetkého veľa škodí*. Inak tomu nie je ani pri plánovaní.

Plánujeme počas celého životného cyklu vývoja softvérového produktu. Dôležité však je uvedomiť si, kedy a do akej hĺbky a podrobností je potrebné zachádzať. Na začiatku vývoja sú v softvérovom projekte príliš veľké podrobnosti absolútne nepotrebné a nežiaduce. Tie sa začnú konkretizovať až neskôr, v pokročilejšej fáze vývoja.

Myslím si, že je dosť dôležité, aby sme na začiatku plánovania svoju energiu nepremrhali zachádzaním do prílišných detailov, ale aby sme sa skôr sústredili na vytvorenie komplexného, aj keď len hrubého plánu. Podrobnejšie a detailnejšie plánovanie by sme mali robiť počas celého trvania nášho projektu, avšak len na určitý, nie veľmi dlhý časový úsek vopred – v horizonte maximálne niekoľkých dní až týždňov.

Dovolím si tvrdiť, že úspešnosť veľmi detailného plánovania hneď v začiatkoch projektu je takmer porovnateľná s úspešnosťou „neplánového vývoja.“ A tá nie je príliš vysoká...

Plánovať dobehnutie zameškaného na neskôr

Plány vytvárame na to, aby sme sa mohli podľa nich riadiť. Našou snahou je pridržať sa ich.

Dovolím si povedať, že takmer každý plán sa poruší a prekročia sa stanovené termíny. Ale veď na to predsa plány sú... Ak by sme nemali plán, ani by sme o tom nevedeli, že sme s niečím pozadu a že „nestíhame termíny.“ Ak však máme plán, dokážeme celkom ľahko, rýchlo a hlavne jednoznačne určiť „inkriminovaný zlom“, keď sme sa dostali mimo rámec plánu. Takúto situáciu je potrebné *riešiť* – okamžite.

Častou chybou v plánovaní býva predpoklad, že stratený čas získame neskôr, a preto sa danou, novo vzniknutou situáciou až tak vážne nezaobráame. Takýto prístup nie je veľmi správny, pretože na neskôr máme naplánované iné činnosti. Výsledkom je potom zakomponovanie takejto „dobiehajúcej úlohy“ medzi už naplánované, čo však vo väčšine prípadov spôsobí aj ich zdržanie, a teda ešte väčšie zdržanie celého projektu.

Nepoučiť sa z chýb minulého plánovania

Všetko zlé je na niečo dobré. Každý človek robí chyby. Má však možnosť poučiť sa z nich. Tak prečo ju nevyužiť?

Ak by sme chyby nerobili, nemali by sme sa z čoho poučiť. To však neznamená, že máme robiť chyby naschvál a cielene. Len je potrebné sa z týchto chýb poučiť, a nabudúce – pri „ďalšom pokuse“ ich už nezopakovať.

To je napríklad aj jeden z dôvodov, prečo je potrebné si okrem plánu vytvárať aj projektový denník – aby sme si mohli zaznamenávať aj chyby, ktorých sme sa dopustili. Pretože to, čo je niekde zaznamenané, sa tak ľahko nestratí, nezabudneme na to. Keď sa však budeme spoliehať len na to, čo si „určite zapamätáme“, opak bude pravdou. Zopakujeme presne tie isté chyby (možno nie všetky, ale nejaké určite) ako aj pri plánovaní v minulých softvérových projektoch, keď sme si hovorili: „Nabudúce sa mi to už nemôže stať!“

Za každým plánom sú skrytí ľudia

Na pozadí každého plánu sú ľudia. Ľudia, ktorí plán vytvárali a aj ľudia, ktorých sa plán týka – ktorí ho budú realizovať a snažiť sa naplňať. Tvorba plánov v softvérovom projekte (samozrejme aj v iných odvetviach) by mala brať ohľad aj na ľudí. Na ich potreby, skúsenosti a v nemalej miere aj na ich nedostatky.

Každý dobrý plán musí počítať s určitým rizikom – ľudským faktorom. Pretože to ako bude prebiehať plnenie plánov závisí predovšetkým od ľudí a ich schopností.

Každý plán sa dá rozložiť na menšie časti, a v neposlednom rade, na konkrétne plány jednotlivcov, tzn. členov tímu. Je veľmi dôležité, aby komplexný tímový plán bral do úvahy aj plány samotných členov tímu. Aby sa nestalo, že jeden člen tímu bude maximálne preťažený, pretože bude mať pridelené množstvo úloh, zatiaľ čo druhí „nebudú mať čo robiť.“

Ako som už spomínal, od realizácie čiastkových plánov jednotlivcov závisí plnenie celkového plánu. Každý je sám sebe pán... To môže niekedy spôsobovať nemalé problémy. Sám dobre viem, aký obrovský problém je nezvládanie osobného časového manažmentu. Aj ja s ním mávam občas problém. Je preto podľa mňa výhodné naučiť členov tímu ako si majú správne naplánovať svoj osobný deň – svoje záľuby, voľný čas a samozrejme povinnosti voči tímu. Je potrebné ukázať im ako by mal vyzerat' správny osobný plán a dať im rady a tipy na lepšie hospodárenie s časom.

Vo väčších projektoch by možno nebolo na škodu zorganizovať školenia ohľadom tejto témy, prípadne poskytnúť jednotlivým členom tímu nejakú podpornú literatúru. V každom prípade by však bolo obrovskou chybou vyvíjať na ľudí nátlak, aby pracovali efektívnejšie a striktne dodržiavali projektový plán, a pritom im neposkytnúť možnosť a priestor na zdokonalenie sa a na vyriešenie svojich problémov.

Najbližšie máme v pláne...

Od kvality plánu závisí mnoho vecí, tak prečo nevytvárať dobré plány, ktoré nám môžu len a len pomôcť? Chyby, ktoré sa vo všeobecnosti vyskytujú v plánoch dokážeme odstrániť, a z obyčajných, prípadne zlých plánov tak vytvoríme kvalitnejšie a lepšie plány – dobré plány. A predsa s dobrými plánmi nemôžeme nič pokaziť, práve naopak – len dosiahnuť lepší výsledok ako ostatní.

Naším najbližším plánom by teraz malo byť vytvorenie dobrého plánu...

Použitá literatúra

1. Armour, G. P.: The Business of Software: To Plan, Two Plans. *Communications of the ACM*, Vol. 48, No. 9, (Sept 2005), 15-19.
2. Bieliková, M.: *Manažment v softvérovom inžinierstve*, 1999.
3. Collins-Cope, M.: *Planning to be agile? A Discussion of How to Plan Agile, Iterative and Incremental Developments*, [online], Rev. 4, Dec 2002, [citované 25.10.2007], <http://www.ratio.co.uk/whitepaper_12.pdf>.
4. McConell, S.: The Nine Deadly Sins of Project Planning, *IEEE Software*, Vol. 18, No. 5 (Sept/Oct 2001), 5-7.
5. Rettig M., Simons, G.: A Project Planning and Development Process for Small Teams, *Communications of the ACM*, Vol. 36, No. 10, (Oct 1993), 45-55.

Annotation

Is every good plan really so good?

For a software project to be successful, a well made plan is necessary. There are various methods used to create plans, but problems can occur when using any of them, that will force us to change our plans. A problem does not emerge by itself, without a reason, on the contrary... One of the reasons are mistakes, which we tend to make during the process of planning. And

it's these mistakes I shall address in the following paper. Together we will look at 9 basic mistakes of planning in software projecting, their causes and consequences. Furthermore, I shall attempt to emphasise the human factor in the process of planning – human as an inseparable part of planning and developing a software project. Human, on whom everything stands and falls, on whom everything depends, for no man, like no plan, is perfect.