

AKO NAPLÁNOVAŤ (NE)ÚSPEŠNÝ PROJEKT

To najcennejšie čo môžete urobiť je chyba, pretože z dokonalosti sa človek nič nenaučí.

Lubomír Elko

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

Abstrakt. Vývoj softvéru je zložitý a časovo náročný proces, ktorý vyžaduje mať veľmi dobre zvládnutú každú svoju časť. Pre potreby efektívneho manažovania a úspešného zvládnutia projektu je však potrebný hlavne kvalitne vypracovaný plán. Hoci existujú softvérové projekty rôzneho charakteru, tak v každej skupine sa nájde projekt, ktorý mešká, či dokonca neskončí úspešne. Ako je to možné? Prečo je to tak? A ako sa vlastne dá v maximálnej možnej miere vyhnúť zlyhaniu? Táto esej zodpovedá práve tieto otázky. Poukazuje na hlavné chyby pri plánovaní softvérových projektov a navrhuje, ako sa im vyhnúť alebo ich aspoň minimalizovať. Taktiež sa zaoberá, ako správne tvoriť plán softvérového projektu, aby viedol k úspešnému koncu.

Kľúčové slová: plánovanie projektov, vývoj softvéru, softvérové inžinierstvo

Úvod

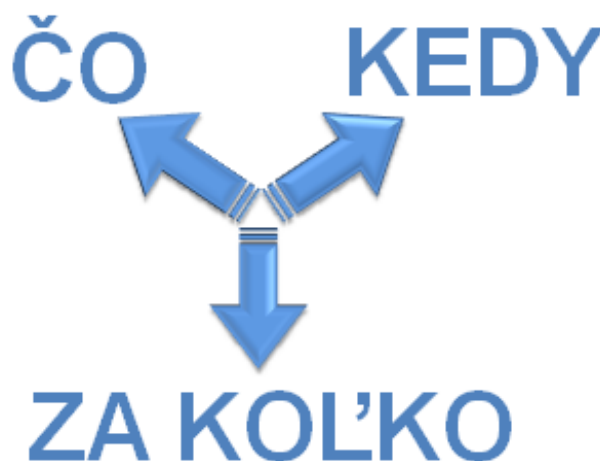
Vývoj softvéru sa skladá z niekoľkých etáp, pričom každá je nám dobre známa aj z bežného života. Prvou etapou je analýza, s ktorou sa dennodenne stretávame napríklad pri rozmýšľaní, čo si zajtra oblečieme do práce, školy alebo kamkoľvek inam. Musíme zohľadniť najmä ročné obdobie, predpokladané počasie na tento deň ale aj špecifiká destinácie kam smerujeme. Samozrejme však máme aj vlastné predstavy o tom, ako chceme vyzerieť, v čom sa budeme cítiť dobre, a to najdôležitejšie, aké šaty vôbec máme k dispozícii. Dalo by sa to prirovnať k špecifikácii požiadaviek a cieľov. Až potom si môžeme navrhnúť akýsi variant toho, čo by sme si mohli obliecť. Nasleduje časť

implementácie, pri ktorej si všetko pripravíme. Môžeme si to predstaviť ako vyžehlenie šiat, čistenie topánok, či mužom známe hľadanie obľúbenej košeľe alebo kravaty. Nadišiel deň s veľkým D. Pripravení sa postavíme pred zrkadlo kvôli poslednej kontrole vzhľadu a vyberieme sa do nášho cieľa. Dá sa povedať, že podstupujeme akési akceptačné testy zákazníka a prichádzame do ostrej prevádzky, pri ktorej je samozrejme dôležité mať dobre zvládnutú aj časť podpory a údržby. V bežnom živote teda mať napríklad po ruke náhradnú košeľu, pokiaľ sa jedná o významnú udalosť, alebo dáždík pri nevyspytateľnom počasí. To všetko môže ale nemusí smerovať k úspešnému koncu, pretože v ktorejkoľvek etape môže nastať nepredvídateľná situácia, ktorá zabrzdí alebo dokonca úplne zastaví tento proces.

Hoci si to teda možno neuvedomujeme, tak aj v každodennom živote sa stretávame s problémami podobnými problémom, s ktorými sa musia vysporiadať plánovači softvérových projektov. Aby sme teda docielili úspešný koniec, potrebujeme mať veľmi dobre prepracovaný plán projektu.

Úspešnosť projektu a plánovanie

Podľa [6] znamená úspešnosť projektu splnenie cieľa v troch dimenziách: vecná (ČO sa má urobiť), časová (KEDY sa to má urobiť) a nákladová (ZA KOLKO sa to má urobiť). Tomu, že projekt má trojdimenzionálny cieľ, hovoríme, že projekt je vždy riadený tzv. *trojimperatívom projektu* (pozri Obr. 1). Môžeme to považovať za dobrú definíciu nakoľko tieto tri otázky sú asi najdôležitejšie pri preberaní produktu (teda toho, ČO sa urobilo). Zákazník totiž nechce aby to, čo si objednal, meškalo, predražilo sa alebo nebodaj nesplnilo jeho očakávania. Práve preto sa už pred podpisom zmluvy o začatí projektu analyzujú tieto tri dimenzie a tvorí sa plán projektu. Vynárajú sa však tieto otázky: Do akej miery je potrebné ich naplniť? Môžeme splniť jednu dimenziu cieľa nad rámec požiadaviek, pričom však nesplníme na 100% inú dimenziu? Odpovede sú pri každom projekte iné. Každý totiž má svoje špecifiká a presne (alebo voľnejšie) určené pravidlá. Tvorba plánu na splnenie zadania teda nie je jednoduchá.



Obr. 1. Trojimperatív projektu [6].

Podľa [1] nastáva plánovanie po prvý krát po inicializácii a jeho dôvodom je hlavne zníženie neistoty, neurčitosti a nepredvídateľnosti. Podľa mňa však ide aj o zvýšenie výkonnosti tímu a v konečnom dôsledku teda aj šetrenie času. Keď má totiž každý v tíme určenú svoju úlohu, vie, čo má robiť a dokedy to má urobiť, tak sa podľa toho môže zariadiť. Nadriadení tiež vidia, že keď sa niekomu nedarí plniť časový plán, tak zrejme bude potrebné prijať do tímu ďalšieho člena alebo neefektívneho člena vymeniť. Keďže však plánovanie pozostáva z viacerých procesov ako sú napríklad manažment rozsahu (angl. scope), nákladov, rozvrhu, integrácie atď., tak manažéri môžu kontrolovať a sledovať priebeh každej časti projektu a vyvodzovať z toho dôsledky.

Nie všetko sa však dá predvídať a kontrolovať. Projektoví manažéri sa najviac obávajú práve prekvapení, teda toho, s čím nerátali. Prekvapenie nemusí prísť len zo strany pracovníkov, ale aj zo strany zákazníka. Ten si môže zmyslieť, že na základe nejakých impulzov prostredia je potrebné upraviť ciele. Dnes sa svet borí v problémoch globálnej finančnej a hospodárskej krízy, a tak takýmto impulzom často býva potreba znížiť náklady naplánované minúť na projekte. Výsledkom nemusí byť len skreslenie rozpočtu alebo pozastavenie na istý časový úsek ale dokonca aj úplne zrušenie projektu.

Čo by teda bolo najlepšie? Predvídať nepredvídateľné! Keďže to však nie je možné, tak treba hlavne minimalizovať dôsledky predvídateľných problémov a vyvarovať sa chýb, ktoré pri plánovaní môžeme spraviť.

9 smrteľných hriechov plánovania

Podľa [3] je vo všeobecnosti zlé plánovanie projektu jedným z hlavných zdrojov problémov. Autor v článku opisuje deväť z najnebezpečnejších hriechov pri plánovaní projektu. Ja ich tu uvádzam spolu s mojimi názormi a komentármi.

Prvým hriechom pri plánovaní je to, že sa jednoducho neplánuje vôbec. Tomuto hriechu sa však dá veľmi ľahko predísť. Podľa mňa sa to často ospravedlňuje tým, že na to nie sú peniaze alebo nie je čas. Problémom však je, že ešte viac peňazí alebo času sa potom minie na riešenie problémov, ktoré nemuseli vzniknúť pokiaľ by existoval nejaký plán.

Druhým je že sa plánuje nedostatočne. Steve McConnell poukazuje na to, že niektoré projektové plány sú vytvorené s predpokladmi, že nikto z tímu neochorie, nezúčastní sa školenia, nepôjde na dovolenku, alebo tím neopustí. Faktom tiež je, že plány vytvorené na základe nereálnych predpokladov, ako sú tieto, môžu mať pre projekt katastrofálne následky. Do tejto skupiny hriechov patria podľa môjho názoru aj chyby pri predvídaní, koľko času bude potrebné vyčleniť na jednotlivé etapy alebo plánovanie typu „Toto sa naplánuje neskôr“.

Nedostatočné plánovanie rizík je podobné druhému hriechu, no tu zohráva hlavnú úlohu otázka, či sme dopredu naplánovali ako sa vyhneme zlyhaniu. Je totiž veľmi nebezpečné nevyjadriť sa v projekte o možných rizikách a ich riešeníach. Tuto by som podotkol, že dobrý plánovač musí mať schopnosť predvídať, čo všetko môže nastať. Musí analyzovať všetky možnosti a vyvodiť z nich dôsledky pre projekt. A to hlavne tie s negatívnym vplyvom.

Nebezpečným prehreškom je aj používanie toho istého plánu na všetky projekty. Aj keď mnohé prvky jedného plánu môžu byť rovnaké ako pri inom projekte, tak je potrebné veľmi dôkladne premyslieť, do akej miery prvky z predchádzajúceho plánu stále platia pre

nový projekt. Už aj ja som stretol pri štúdiu, že projekt na jednom predmete bol veľmi podobný projektu na inom predmete. Použitie starého plánu a dokumentácie je však veľmi nebezpečné. Môže sa totiž stať, že si zmýlime ciele a implementovať budeme aj to, čo nie je potrebné alebo naopak, neimplementujeme to, čo treba.

Podobným hriechom je aj uplatňovanie jedného pred pripraveného plánu. Takýto plán síce môže pomôcť vyhnúť sa prvému a druhému hriechu, no nezohľadňuje jedinečné potreby nového projektu. Takéto plány by som nazval „rýchlo kvasené“ a podľa mňa nie sú kvalitné. Pri plánovaní totiž treba čas na dôkladné premyslenie každej drobnosti. V kontexte nového projektu totiž môže byť najdôležitejšia práve drobnosť, ktorú sme pri minulých projektoch nezohľadňovali.

Najčastejším hriechom je asi odchýlenie sa od projektovej reality. Dôjde k nemu najčastejšie vtedy, keď sa na začiatku projektu vytvorí plán, no potom sa, obrazne povedané, položí na policu a nechá sa na neho sadať prach na zostávajúcu časť projektu. Už v úvode som písal o problémoch v čase globálnej hospodárskej krízy. Zmeny v podmienkach projektu teda treba okamžite zohľadniť v pláne. V žiadnom prípade netreba nenechávať plán nemenný po celú dobu projektu.

Problémom je aj keď sa všetky detaily plánujú príliš skoro. Pokúšajú sa o to najmä skúsení projektanti. Softvérový projekt sa však skladá zo stále sa rozvíjajúceho množstva rozhodnutí a každá fáza projektu vytvára závislosti pre budúce rozhodovania. Tuto sa mi zdá, že to trochu odporuje druhému hriechu, no je potrebné si uvedomiť, že každý softvérový projekt je iný a takisto, že všetkého veľa škodí. Správny projektant by mal mať triezvy pohľad na problémy a mal by vedieť usúdiť, čo sa momentálne nedá naplánovať.

Ďalším hriechom je plánovanie, že strata sa doženie neskôr. Pri projektoch, ktoré by mohli zaostávať za plánom, sa niekedy plánuje doháňanie prípadnej straty. Tento hriech sa vytvára v domnienke, že na konci projektu už všetci členovia tímu budú vedieť podrobnosti o projekte, budú ľahšie riešiť problémy a tak teda budú aj rýchlejšie pracovať. Avšak to je chyba! Stačí keď pripomeniem fakt známy ako neviditeľnosť softvéru. Tento jav nastáva podľa [2] vtedy, keď je projekt už zdanlivo takmer ukončený, ale potrebná práca presahuje množstvo tej vykonanej. Dohnanie straty je práve v takýchto prípadoch nereálne.

Najvražednejším hriechom zo všetkých je neponaučenie sa z predchádzajúcich chýb! Softvérové projekty môžu trvať veľmi dlho a faktom je aj to, že plynutím času človek zabúda, takže sa môže ľahko stať, že pri plánovaní spravíme opäť tú istú chybu ako v minulosti. Podľa mňa v tomto hriechu zohráva istú rolu aj fakt, že niektorí ľudia si nedokážu priznať chybu. Tvrdohlavo si idú za svojim cieľom, pričom nemajú ochotu sa nechať poučovať ostatnými.

Čomu sa ešte treba vyvarovať?

Podľa môjho názoru je ešte potrebné spomenúť ďalšie hriechy, ktorých sa plánovači softvérových projektov často dopúšťajú. Možno nie sú až také smrteľné ako deväť vyššie spomínaných, no vyvarovaním sa pred nimi určite uľahčíme vývoj softvérového projektu.

Niekedy sa stáva, že plánovači, hlavne tí menej skúsení, nerozumejú celkom práci programátorov alebo testerov a neodhadnú dobre čas potrebný na ich prácu alebo potreby, aké majú. Tí viac skúsení plánovači zase nemusia zohľadňovať ich požiadavky a

kladú potom vysoké nároky. Prirovnal by som to k dvom stranám mince. Isté naplánované kroky do budúcnosti, ktoré môžu uľahčiť prácu plánovačom, môžu na druhej strane zároveň sťažiť prácu pri implementácii. Plán totiž z pohľadu plánovača môže vyzerá ináč ako napr. z pohľadu testera.

V [4] sa ako jeden z problémov udáva situácia behu viacerých úloh. Tá nastáva keď musí jeden človek pracovať na viacerých projektoch. V podstate je jedno na akej pozícii vystupuje, pretože sústredenie sa na obe projekty je veľmi namáhavé a hoci by to nemalo byť možné, tak sa často stáva, že problémy z jedného projektu sa presúvajú aj na druhý. Neveríte? Predstavte si situáciu, že vývojárovi sa nedarí dodržiavať plán prvého projektu a tým pádom na tomto projekte musí pracovať rýchlejšie a venovať mu viac času. To sa však veľmi rýchlo odzrkadlí na plnení plánu druhého projektu, lebo na neho bude mať menej času, vývojár bude viac unavený, bude pracovať vo väčšom strese a v takýchto podmienkach bude zákonite robiť aj viac chýb. Je preto potrebné veľmi dobre zvážiť použitie jedného človeka na viacerých projektoch.

Častou chybou je aj prílišný optimizmus. Na základe toho sa zvolí tzv. optimistický plán. Takéto plány sa síce tvoria, no neodporúča sa ich používať pri projektoch. Mali by mať skôr len charakter jedného variantu softvérového projektu. Vhodné je teda vypracovať aj realistický a pesimistický plán. Nastáva teraz otázka, ktorý plán je ten správny? Ktorý zvolíte? Po prečítaní smrteľných hriechov, ktoré som spomínal vyššie, jasne vidíme, že ako prvý vylúčime optimistický plán. Keď vezmeme do úvahy nutnosť splniť projekt v troch dimenziách (teda obsah, čas a náklady), ktoré sú často určené zákazníkom, tak usúdime, že pesimistický plán asi niektorú požiadavku nesplní. Realistický plán je dobrý, no vždy by sme mali mať na pamäti, že musíme počítať aj s prekvapeniami. Ideálne je teda podľa môjho názoru zvolenie realistického plánu s tým, že do neho zahrnieme isté mierne pesimistické prognózy a naplánujeme ich riešenie, pokiaľ by nastali.

Ako teda naplánovať úspešný projekt malého tímu?

V [5] autor popisuje plánovanie projektu a vývojového procesu malého tímu. Okrem vedúceho by v tíme mal byť aj aspoň jeden manažér, ktorý vedúcemu pomáha pri rozhodovaní ako rozdeľovať zdroje alebo ako odstraňovať problémy, ktoré sa vyskytnú. Len vedúci však zodpovedá za výsledky, ktoré vyprodukuje tím. Vďaka tomu sa zvyšok tímu nemusí starať o firemnú politiku a môže sa plne sústrediť na svoju prácu. Toto je podľa mňa veľmi dôležité. Takisto by som privítal, keby mal každý člen tímu len jedného priameho nadriadeného, ktorému sa má zodpovedať. Tým by sa totiž dalo vyhnúť situácii, že dvaja nadriadení by mali iný pohľad na jeho prácu a požadované výsledky. Pracovník by bol teda v rozpakoch, čo má vlastne robiť.

Autor ďalej uvádza istú osnovu dobrého plánu. Projekt by mal mať jednoznačný názov, ktorý ho odlišuje od ostatných projektov softvérovej firmy. Mali by byť jasne uvedené ciele a účel projektu a zoznam vývojových fáz, ktorými je potrebné prejsť počas cesty k dokončeniu. Tieto fázy treba zoradiť podľa toho, kedy je potrebné ich mať vyriešené a každej fáze určiť časový harmonogram riešenia problémov a rizík, ktoré obsahuje. Poslednú súčasť plánu, ktorú autor spomína, je opis členov tímu spolu s ich rolami. Možno nedopatrením však zabudol na jednu z najdôležitejších častí plánu, a to na

rozpočet softvérového projektu. Bez neho by sme totiž nemali tretiu (nákladovú) dimenziu, ktorá je potrebná pre určenie úspešnosti projektu.

Poslednú radu, ktorú uvediem, je známe porekadlo, že to najcennejšie čo môžete urobiť je chyba, pretože z dokonalosti sa človek nič nenaučí. A to pri plánovaní softvérových projektov platí dvojnásobne. Chyba sa totiž veľmi rýchlo prejaví a často aj zobrazí vo výsledku. Poučiť sa z toho je teda veľmi dôležité!

Záver

Touto esejou som sa snažil poukázať na hlavné chyby pri plánovaní softvérových projektov. Navrhol som tiež ako sa im dá vyhnúť alebo ich aspoň minimalizovať. Esej sa zaoberá aj problematikou ako správne tvoriť plán softvérového projektu aby viedol k úspešnému koncu. Verím, že čitatelia sa nad diskutovanými problémami zamyslia a zoberú si z nich cenné rady, ktoré zúžitkujú pri plánovaní softvérových projektov.

Použitá literatúra

1. Bartoš, P.: *Plánovanie softvérového projektu*. Prednáška z predmetu Manažment projektov softvérových a informačných systémov, FIIT STU v Bratislave. 06.10.2009.
2. Bieliková, M.: *Ako úspešne vyriešiť projekt*. STU Bratislava, 2000. 158 s. ISBN 80-227-1329-5.
3. McConnell, S.: The Nine Deadly Sins of Project Planning. *IEEE Software*, Vol. 18, No. 5 (2001) 5-7.
4. Okoro, S.: *Why Over 90 Percent of All Projects Finish Late*. [cit 2009-10-11]. Dostupné na internete: <<http://www.projectsmart.co.uk/pdf/why-over-90-percent-of-all-projects-finish-late.pdf>>.
5. Rettig, M., Simons, G.: A Project Planning and Development Process for Small Teams. *Communications of the ACM*, Vol. 36, No. 10 (1993) 45-55.
6. Staníček, Z., Hajkr, J.: Řízení projektů zavádění IS do organizací. In: *Datakon*, 2005 [cit. 2009-10-13]. Dostupné na internete: <<http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/reporty/datakon2005tutorial.pdf>>.

Annotation

How to plan (un)successful project

Software development is complex and time-consuming process that requires a very well controlled each of its parts. For the purposes of managing the efficient and successful management of the project is needed especially well-developed plan. Although there are software projects of different nature, so in each group will find a project that is delayed or even completed successfully. How is this possible? Why is this so? And how indeed can be as much as possible to avoid failure? This corresponds exactly to the essay questions. It refers to major errors in planning software projects, and suggests how to avoid them or at least minimized. It also addresses how to create a plan software project that led to a successful end.