

PLÁNOVANIE V MALOM

Mať dobrý plán, to je základ úspešnosti.

Adrián Feješ

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
fej0089[zavináč]gmail[.]com

Abstrakt. *Keď človek narazí na nejaký zložitejší problém, ktorý treba vyriešiť, prvé čo robí je, že začne rozmýšľať o možnostiach riešenia. Ak nájde vhodné riešenie tak sa začne sústrediť na spôsob jeho realizácie. Prejde riešenie krok po kroku, pričom uvažuje o potrebných zdrojoch. Na softvérový projekt je tiež možné pozeráť ako na problém, a začať navrhovať postupnosť krokov a definovať potrebné zdroje, čím sa postupne vytvára plán projektu. Samozrejme čím väčší je rozsah projektu a čím viac ľudí sa zúčastňuje na jeho riešení, tým bude plán projektu zložitejší a stúpa jeho dôležitosť. Ako ovplyvní veľkosť projektu jeho plánovanie? Môžeme pri plánovaní malých projektov postupovať rovnako ako pri plánovaní tých väčších? Hľadanie odpovedí na tieto a podobné otázky je predmetom tejto eseje. Pričom pre nájdenie odpovedí je nutné priblížiť aj význam pojmu malý softvérový projekt.*

Kľúčové slová: *plánovanie, malé softvérové projekty, plánovanie softvérových projektov*

Úvod

Plánovanie by malo byť bežnou súčasťou nášho života a nielen privilegium projektových manažérov. Vopred plánovať svoj život by mal každý z nás. Hlavnou motiváciou pre plánovanie je úspešné zvládnutie každodenných problémov. To isté platí aj pre softvérové projekty, kde hnacou silou je taktiež úspešnosť. Pri plánovaní softvérových projektov sa dajú využiť všeobecné znalosti a techniky manažmentu projektov. Pre vytváranie úspešného plánu nestačí tie znalosti poznať, musíme ich aj prispôbovať špecifikám softvérových projektov. Ďalšia vec, ktorú treba zohľadniť pri plánovaní projektov je ich veľkosť. Veľkosť projektu nie je niečo, čo sa dá jednoznačne merať a to obzvlášť platí pre

2 Adrián Feješ

softvérové projekty. Je preto zaujímavé vymedziť pojem malý softvérový projekt a zaoberať sa jeho plánovaním.

Čo sa skrýva pod pojmom plánovanie?

Teraz sa skúsme pozerať na plánovanie z odborného hľadiska. Plánovaním, ako procesom manažmentu softvérových projektov sa zaoberá veľa literatúr a tak vzniklo viacero definícií. Za najvýstižnejšiu považujem definíciu podľa ktorej plánovanie je: „*Vytvorenie a udržiavanie schémy na zabezpečenie chodu projektu.*“ [1]. Táto jedna veta perfektne vystihuje dôležitosť plánovania. Podľa nej plán je nevyhnutnou súčasťou projektu, ktorá zabezpečuje jeho úspešné vykonávanie. Ďalšia vec, ktorú si môžeme všimnúť je, že plán nestačí vytvoriť, ale je nutné ho aj často aktualizovať. Treba mať na pamäti, že neaktuálny plán je veľmi nebezpečný. Stojí za povšimnutie aj to, že citovaná definícia nehovorí nič o veľkosti projektu. Môže ísť buď o projekt, na ktorom sa zúčastňujú stovky ľudí, alebo o projekt, ktorý zamestná len 5-6 ľudí.

Malý softvérový projekt

Keď som začal hľadať informácie o plánovaní malých softvérových projektov narazil som na niekoľko problémov. Prvým a asi najpodstatnejším z nich bolo zistenie, že meranie veľkosti projektov vôbec nie je jednoduché. Neexistujú presné kritéria na určenie toho, či uvažovaný projekt je malý, stredný alebo veľký. Samozrejme existujú rôzne odporúčania, ktoré poskytujú určité prostriedky na meranie veľkosti projektov. Jedno z takýchto odporúčaní zaznelo aj v publikácii, zaoberajúcej sa procesom vývoja malých softvérových projektov [3]. Tu sú štyri faktory, ktoré podľa nej môžu ovplyvniť veľkosť projektov:

- Veľkosť organizácie realizujúcej projekt
- Zložitosť projektu
- Atribúty kvality
- Interakcie ľudí

Pozrieme sa bližšie na uvedené faktory. Po skúmaní, ako veľkosť organizácie môže ovplyvniť veľkosť projektu som skonštatoval nasledujúce: Veľkosť organizácie môže mať dvojaký vplyv na veľkosť projektu, pozitívny a negatívny. Pre veľkú organizáciu, ktorá má dobre definované procesy, má dostatok kvalifikovaných ľudí a skúsenosti, sa môže aj pomerne zložitý projekt zdať ako malý. Na druhej strane veľká organizácia so zložitou organizačnou štruktúrou môže aj malý projekt zaťažiť administratívou a zložitým spôsobom riadenia a robiť z neho veľký projekt.

Ďalším, a podľa mňa veľmi dôležitým faktorom je interakcia alebo komunikácia ľudí. Je zrejmé, že projekt, ktorý zamestná desiatky až stovky ľudí si vyžaduje oveľa viac úsilia na zabezpečenie komunikácie, než projekt ktorý zamestná 4 – 5 ľudí. Pričom si treba uvedomiť, že pri komunikácii nejde len o kvantitu ale aj o kvalitu. Komunikácia medzi 5 ľuďmi, ktorí sa dobre poznajú a sedia v jednej miestnosti je zvyčajne oveľa efektívnejšia, čo vo významnej miere zefektívni aj vykonávanie celého projektu.

Vplyv veľkosti na plánovanie

V dnešnej dobe globalizácie je na mieste otázka, prečo vznikajú malé projekty a prečo je dôležité sa nimi zaoberať. Je vhodné si uvedomiť, že menšie projekty sú často súčasťou väčších projektov, a majú vlastný životný cyklus. Ďalšou motiváciou pre realizáciu menších projektov je experimentovanie s novými technológiami, resp. s novými procesmi vývoja. Malé projekty sú vhodné aj na tréning nových projektových manažérov.

Vidíme, že malé softvérové projekty majú potenciál, a preto má zmysel zaoberať sa jeho plánovaním. Malé projekty sa vyznačujú menším rozsahom, menším počtom ľudí resp. kratším trvaním. Všetky tieto skutočnosti majú vplyv na ich plánovanie. Ak sa pozrieme na jednotlivé podprocesy plánovania, tak vidíme, že ich zložitosť priamo závisí od veľkosti projektu. Definovanie rozsahu projektu je tým jednoduchšie, čím je projekt rozsahu menší. Pridelenie zdrojov k jednotlivým úlohám je tým jednoduchšie čím je menej úloh. A takto by sme mohli ďalej pokračovať. Naozaj je to také jednoduché? Odpoveď na otázku je samozrejme nie. Stačí si uvedomiť, že ak má projekt menší rozsah, tak zvyčajne má k dispozícii menej zdrojov resp. aj tie dostupné zdroje sú obmedzené. S menším rozsahom sa zvyčajne zredukuje aj čas, ktorý je k dispozícii na realizáciu projektu. To znamená, že pri plánovaní malých projektov síce máme na definovanie a naplánovanie menej úloh, ale súčasne máme k dispozícii menej zdrojov a menej času. Menej času znamená, že aj samotné plánovanie musíme realizovať rýchlejšie a efektívnejšie. Teraz je už otázne, či je vôbec plánovanie malých softvérových projektov jednoduchšie ako plánovanie tých väčších.

Ako na to?

Už vieme čo je to plánovanie, a aké procesy v sebe zahŕňa. Po prečítaní predchádzajúcich častí eseje už by sme mali mať predstavu aj o tom, čo charakterizuje malé projekty. Teraz sa pozrieme bližšie na proces plánovania malých softvérových projektov.

Aj keď bolo spomenuté, že plánovanie závisí od veľkosti projektov, samozrejme plánovanie, ako proces manažmentu projektov vždy sleduje rovnaký cieľ. Tým cieľom je vytvorenie základu pre riadenie a sledovanie projektu. To znamená, že vytvorené plány majú podobný obsah, akurát sa líšia v rozsahu. Pod podobným obsahom sa rozumie, že obidva plány v nejakej podobe obsahujú základné stavebné prvky plánu, akými sú rozsah, rozvrh, rozpočet atď. Samozrejme niektoré časti plánov, najmä tie, ktoré viac závisia od rozsahu projektov budú odlišné resp. môžu chýbať v niektorých plánoch. Čím sa plánovanie malých projektov líši od plánovania tých väčších je rozsah a spôsob vykonávania procesov plánovania.

Plánovanie závisí aj od spôsobu vývoja, ktorý si daný tím zvolil na realizáciu projektu. Vývoj softvéru môže prebiehať podľa jedného zo známych modelov životného cyklu softvéru, ako napr. klasický vodopádový model, špirálový model alebo moderné agilné modely. Vodopádový model vývoja asi pozná každý. Aj keď sa v dnešnej dobe už veľmi nepoužíva mal veľký význam a od neho sa postupne odvíjali modernejšie a úspešnejšie modely životného cyklu softvéru. Práve kvôli jeho nevýhodám sa teraz ním podrobnejšie nebudeme zaoberať pri definovaní procesu plánovania malých softvérových projektov.

Oveľa úspešnejším modelom vývoja je špirálový model. Princípom tohto modelu je, že vývoj softvéru sa rozdelí na etapy, ktoré pozostávajú z rôznych „modulov“. Menšie projekty môžu byť súčasťou väčších projektov. V tomto prípade plán pre menší projekt sa môže vytvárať ako podrobnejší plán jednej etapy toho väčšieho projektu. Práve v takýchto situáciách je vhodné aplikovať špirálový model. Ďalšou možnosťou je, že malý projekt sa považuje za samostatný. Vo väčšine prípadov tieto projekty sa zameriavajú pre špeciálnu oblasť, ktorá je pre členov tímu neznáma. V takomto prípade plán treba prispôbiť tak, aby zohľadňoval počiatočné ťažkosti súvisiace s neznámym prostredím. Aplikáciou špirálového modelu sa projekt rozdelí na etapy, ktoré sa vykonávajú v iteráciách. Každou iteráciou sa tím viac porozumie problémovej oblasti a vytvára sa tak predpoklad na úspešnejšie zvládnutie ďalších etáp.

To, že špirálový model je vhodný na plánovanie a vývoj malých softvérových projektov dokazuje aj publikácia [2], kde je navrhnutý proces plánovania na malé softvérové projekty. Proces je založený na špirálovom modeli. Podľa neho sa projekt rozloží na etapy, ktoré pozostávajú z rôznych fáz. Nasledujúce 4 fázy boli definované: maketovanie, prototypovanie, vylepšovanie a udržovanie.

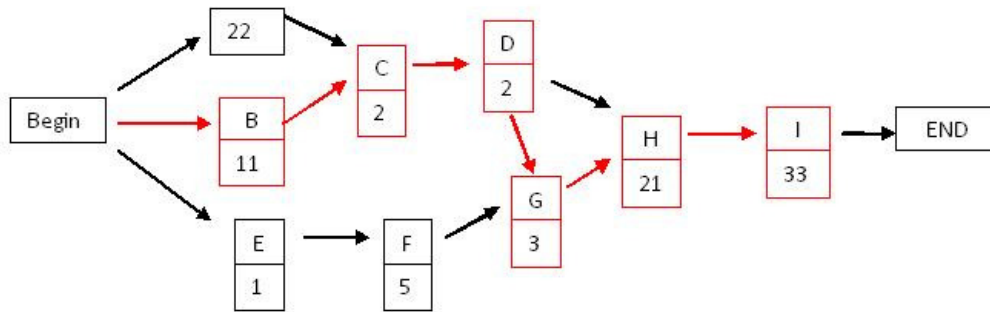
Využitie známych techník v malom projekte

Teraz sa podrobnejšie pozrieme na niektoré podprocesy plánovania, konkrétne na vytvorenie rozvrhu a na odhad nákladov, pričom sa sústredíme na malé projekty. Práve tieto dva procesy by sme mali vykonávať pri plánovaní každého projektu.

Pre vytvorenie rozvrhu existuje veľa techník, my sa teraz budeme zaoberať metódou kritickej cesty (anglicky CPM – Critical Path Method) (Obr.1.). Medzi hlavné výhody tejto techniky podľa mňa patrí: prehľadná grafická reprezentácia projektu, umožňuje odhadnúť čas potrebný na realizáciu projektu a pomôže pri odhaľovaní kritických úloh. Pri hľadaní kritickej cesty zvyčajne postupujeme nasledovne:

1. Identifikujeme jednotlivé činnosti
2. Zoradíme identifikované činnosti a nakreslíme sieť činností
3. Určíme čas dokončenia jednotlivých činností
4. Následne identifikujeme kritickú cestu ako najdlhšiu cestu v sieti

Ak vyššie opísanú postupnosť krokov vykonáme v prostredí malého projektu zistíme niekoľko zaujímavých skutočností. Počet identifikovaných činností bude pomerne malý. Po vytvorení siete činností zistíme, že výsledná sieť je síce prehľadná ale veľmi jednoduchá. Malý počet činností spôsobuje, že medzi jednotlivými činnosťami sa vytvára menej vzťahov a závislostí. Je na mieste otázka, či takáto jednoduchá sieť činností nám poskytuje dostatok informácií a či sa oplatí investovať čas do jej vytvorenia? Podľa môjho názoru, aplikovanie metódy kritickej cesty má význam aj pre menšie a jednoduchšie projekty, už len z tohto dôvodu, že menší počet činností vytvára predpoklad na jej rýchlejšie vytváranie. Navyše si myslím, že každý nástroj, ktorý nás donúti rozmýšľať o probléme prispeje k jeho úspešnému zvládnutiu.



Obr.1. Metóda kritickej cesty [5].

Pri definovaní charakteristík malých projektov som spomenul, že majú tendenciu zamerať sa na špeciálne problémové oblasti. To väčšinou spôsobuje, že odhadnúť trvanie jednotlivých činností je ťažšie. Táto skutočnosť je veľmi dôležitá pri vytváraní siete činností a ovplyvňuje úspešnosť odhadu kritickej cesty. Práve preto si myslím, že metóda kritickej cesty je vhodná najmä na komplexnejšie projekty, ktorých trvanie činností vieme presne odhadnúť.

Takisto ako pri plánovaní veľkých softvérových projektov aj pri plánovaní tých menších dôležitú úlohu hrá odhadovanie. Na odhad nákladov projektu existuje viacero techník a metód, my sa teraz sústreďíme na model COCOMO 2. Tento model umožňuje odhadnúť úsilie, potrebné na realizáciu projektu. Odhadnuté úsilie sa uvádza v človeko - mesiacoch. Ako základ na meranie úsilia využíva hlavne veľkosť projektu, a preto je zaujímavé pozrieť sa na vlastnosti tohto modelu v prostredí malých softvérových projektoch. Pri tomto modeli sa uvažuje nasledujúcich 5 faktorov rozsahu (anglicky Scale Drivers):

- Precedens (anglicky Precedentedness)
- Pružnosť vývoja (anglicky Development flexibility)
- Architektúra / Riešenie rizík (anglicky Architecture / Risk Resolution)
- Súdržnosť tímu (anglicky Team Cohesion)
- Zrelosť procesu (anglicky Process Maturity) [4]

Skúsme sa teraz pozrieť na jednotlivé faktory rozsahu v kontexte malých softvérových projektov. Ako som už zopárkrát spomenul malé softvérové projekty sa často zameriavajú na špeciálne problémové oblasti, čo vo veľkej miere zníži šancu nájdenia podobného projektu, teda precedensu. Tým, že ide o relatívne malý tím, sa zvyšuje pružnosť vývoja. Čo sa týka architektúry, tak intuitívne očakávame, že projekty s menším rozsahom budú mať jednoduchšiu architektúru, čo ovplyvní aj úspešnosť riešenia rizík. Malý tím, pozostávajúci z ľudí, ktorí sa dobre poznajú a medzi ktorými prebieha neformálna komunikácia má tendenciu ukazovať väčšiu súdržnosť. Čo sa týka posledného faktoru, tak podľa môjho názoru veľkosť projektu priamo neovplyvní zrelosť procesu vývoja.

Záver

Táto esej mala za cieľ priblížiť proces plánovania v kontexte malých softvérových projektov. Treba si uvedomiť, že merať veľkosť softvérového projektu vôbec nie je ľahké a preto nie je ľahké ani špecifikovať postup na plánovanie malých projektov. Jedno je ale isté, plánovanie je potrebné bez ohľadu na veľkosť projektu. Pritom pri plánovaní malých projektov treba brať do úvahy jeho špecifiká, akými sú napr. menší počet úloh, menší počet ľudí, obmedzené zdroje, efektívnejšia komunikácia v tíme. Pri opise aplikovania známych techník v malých softvérových projektoch som sa zameral na vytvorenie rozpočtu a odhad nákladov, pretože ich považujem za najdôležitejšie. Samozrejme veľkosť projektu ovplyvní aj ďalšie procesy plánovania, ktorých analýza z pohľadu malých projektov tiež môže byť zaujímavá.

Použitá literatúra

1. Bieliková, M.: Inicializácia a plánovanie softvérového projektu, 2009, <http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/prednasky/msi03-2009.pdf>
2. Retting, M., Simons, G.: A project planning and development process for small teams. In Communications of the ACM, New York : ACM Press, 1993
3. Russ, L., Melissa, McGregor, D., John: A Software Development Process for Small Projects, IEEE Software, September/October 2000
4. ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf
22.11.2011
5. http://www.angelo.edu/services/project_management/Project_Planning.php
23.11.2011

Annotation

Planning of small software projects

When someone encounters a difficult problem to be solved, first he does is to start thinking about possible solutions. If he finds a suitable solution then he starts focusing on its implementation. A software project can also be seen as a problem and we can define steps and resources that are needed to realize the project, which is a process of creating the project plan. Of course the larger the scope of the project and the more people involved is on its solution, the project plan will be more complex and increases its importance. How does the size of the project influence its planning? Is the approach to planning small project and large project the same? Searching for answers to questions like these is the subject of this essay. To find answers it is necessary to explain the significance of term small software project.