

VYUŽITIE POTENCIÁLU RÔZNYCH PRÍSTUPOV PRI PLÁNOVANÍ PROJEKTU

Plán je cesta, ktorú spoznáme až na jej konci.

Ján Antala

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
jan[zavináč]antala[.]sk

Abstrakt. Plánovanie je jednou z najdôležitejších etáp manažmentu projektu súvisiaceho s tvorbou softvérového produktu. Postupom času vznikajú rôzne techniky plánovania, pomocou ktorých vieme projekt úspešne dokončiť v požadovanom čase a požadovanej kvalite. Každý spôsob má svoje výhody aj nevýhody, a preto je potrebné zvoliť ten správny, alebo si existujúci prispôbiť vlastným potrebám či potrebám projektu. Esej prezentuje rôzne možnosti plánovania. Zamýšľa sa, prečo a kedy by ich bolo vhodné alebo nevhodné použiť a navrhuje lepšie riešenie, ktoré minimalizuje ich nedostatky a zvyšuje realističnosť vytváraného rozvrhu. Navrhnuté riešenie taktiež obsahuje mechanizmy na zabránenie neustálych zmien špecifikácie a s tým spojeným predlžovaním času implementácie produktu. Esej rozmyšľa aj nad možnými rozšíreniami navrhnutého riešenia, pomocou ktorých je možné dospieť k ešte lepším výsledkom, no zároveň môžu so sebou niesť aj určité riziká.

Kľúčové slová: plán, rozvrh, časové odhady, realističnosť, evidencia

Úvod

Mnohokrát sa stáva, že na projekt máme presne pridelený termín. Ako ho však máme dodržať, aj keď zákazník stále nevie, čo presne chce, a stále mení svoje požiadavky? Čo keď vývojár tvrdí, že úloha bude dokončená, keď ju dokončí? Ako vieme povedať, či je daný termín vôbec reálny?

Východiskom z tejto situácie je plán. Ako ale zostaviť plán správne? Najlepší projektoví manažéri vedia, ako zostaviť plán. Vedia, aké má mať projekt medzníky. Rozvrh plánujú špecificky pre úlohy v projekte [3]. Keďže manažér plánovania nie je okamžite majstrom svojho odboru a postupne sa učí zo svojich minulých chýb, ostáva mu na plánovanie použiť nejakú techniku.

Iteratívne plánovanie

Jednou z možností je použiť iteratívne plánovanie, pomocou ktorého je možné obísť rozhodovanie medzi osekáním počtu možností aplikácie a nafaňovaním času implementácie [4]. Toto plánovanie pevne stanovuje dátumy, do kedy sa môžu projektu pridávať nové vlastnosti, do kedy bude prebiehať implementácia, testovanie či uverejnenie produktu na trh. Takýto prístup má samozrejme svoje výhody aj nevýhody.

Hlavnou slabinou je vyvíjanie prehnaneho tlaku na vývojárov a testerov aby implementácia či testovanie prebehlo do stanoveného termínu, čo je jednou s najväčších chýb plánovania [2], nakoľko po čase môžu prestať podávať požadované výkony alebo úplne opustiť projekt. Stále však nevieme povedať, s akou šancou sa nám podarí dohodnutý termín splniť, a v projekte môžu zbytočne ostávať chyby. Taktiež je potrebné rátať s aj rizikami, ktoré môžu vzniknúť pri riešení projektu.

Na druhej strane plánovanie prináša aj pozitíva. Tým najväčším je, že sa projekt zbytočne nenaťahuje keďže presne vieme, akú funkcionality bude obsahovať a ďalšia sa už do vyvíjanej verzie pridávať nebude. Pokiaľ sa však zákazník rozhodne k nejakej úprave funkcionality až po konečnom termíne, tak sa pridá až do ďalšej iterácie produktu a nemožno tak dynamicky reagovať na zmeny.

Čo ak oproti stanoveným medzníkom projekt postupuje príliš rýchlo alebo pomaly? V takom prípade je stále možné pôvodný plán preplánovať.

Plánovanie založené na evidencii

Ak už máme evidenciu z predchádzajúcich projektov môžeme zvoliť plánovanie postavené na nej [5]. Pomocou sledovania času stráveného na jednotlivých úlohách projektu a ich porovnaním s odhadovanými časmi sa dostaneme k zaujímavým hodnotám.

Čo tieto údaje znamenajú? Výsledkom porovnania je relatívna rýchlosť - ako rýchlo boli úlohy dokončené vzhľadom k ich odhadu.

$$\text{relativnarychlost} = \frac{\text{ocakavanycas}}{\text{skutocnycas}} \quad (1)$$

Všetko trvá dlhšie, ako sa pôvodne očakávalo, pretože odhad nezahrňa opravu chýb, čas strávaný na konzultáciách či prestávkach. Priemerná relatívna rýchlosť má však pre všetky úlohy podobné hodnoty menšie ako 1 okolo 0.6 [5].

Na čo získané údaje slúžia? Pomocou získaných údajov vieme simulovať budúcnosť a následne zistiť s akou šancou vieme dodržať dohodnutý termín. Z pôvodne odhadovaného času úlohy a relatívnej rýchlosti získame výsledný odhad času. Postupným priradovaním náhodných relatívnych rýchlostí ku všetkým úlohám v projekte dostaneme

celkom presný odhad trvania celého projektu. Viacnásobným opakovaním simulácie sa výsledné odhady budú blížiť k reálne strávenému času na projekte.

Tab. 1. Získanie pravdepodobného času riešenia projektu.

Očakávaný čas	2h	4h	8h	10h	16h	Celkovo: 70.7h
Relatívna rýchlosť	0.5	0.6	0.6	0.5	0.6	
Pravdepodobný čas riešenia	4h	6.7h	13.3h	20h	26.7h	

Ako je to pri plánovaní v tíme? V skutočnosti má každý vývojár inú relatívnu rýchlosť a tak pri plánovaní a rozdeľovaní úloh je potrebné túto skutočnosť zohľadniť. Je dôležité brať ohľad na to, že niektoré úlohy v projekte môžu na seba nadväzovať a pokiaľ na nich pracujú rôzni ľudia s rôznymi rýchlosťami, môže dochádzať k časovým oknám.

Plánovanie založené na evidencii je celkom dobrou metódou, ako získať skutočnú dĺžku trvania projektu. Stále však vstupuje do hry počiatočný odhad trvania úloh, ktorému sa nevyhneme. Nevýhodou je aj možnosť neustáleho pridávania novej funkcionality zo strany klienta a s tým spojené možné predlžovanie trvania projektu.

Ako si zvolit' správny plán?

Keďže sme splnili prvý krok, ktorým je že chceme projekt naplánovať a nechceme len o výsledku snívať, ostáva ešte zodpovedať, ako ho zvolit' správne. Zadanie vyzerá byť celkom jednoduché. Chceme eliminovať čo najviac nedostatkov a mať plán čo najrealistickejší. Ale je na tom rovnako aj riešenie?

Nebolo by pre lepšiu efektivitu vhodné skombinovať prezentované techniky plánovania? Myslím si, že použitie výhod rôznych riešení a vytvorenie si vlastného spôsobu plánovania založeného na nich vyzerá celkom sľubne. Výsledkom je kombinácia pevne stanovených termínov vychádzajúcich z vlastných štatistík. Nové požiadavky zákazníka vieme ihneď ohodnotiť a pridávať do systému, kým nám na ne ostáva voľný čas.

Samozrejme, že je potrebné mať aj čo najpresnejšie odhady časov od vývojárov. Na minimalizáciu rizika nepresností je vhodná diskusia o jednotlivých úlohách.

Stále si však musíme dávať pozor aj na to, aby sme neplánovali zbytočne dlho a potom nám neostal čas na kompletnú implementáciu projektu. Na vyhnutie sa tejto vlastnosti už plán neexistuje a musí si ju uvedomiť každý manažér plánovania. Výhodnejšia je neskoršia korekcia pôvodného plánu.

Nevýhodou takéhoto riešenia je, že sa nedá použiť na projekt, pokiaľ nemáme evidenciu z minulosti. Nezostáva nám nič iné ako si náhodne zvolit' hodnoty relatívnych rýchlostí pre vývojárov. Podobný problém nastáva aj keď do tímu pribudne niekto nový. Pokiaľ pristúpime k takémuto riešeniu, tak sa zväčšuje neurčitost'. Stále je to ale lepšie riešenie, ako len používať časové odhady od zamestnancov.

Rozšírenia plánu

Ako by sa dalo plánovanie ešte vylepšiť? Samozrejme že žiaden plán nie je úplne dokonalý a je potrebné ho upraviť vlastným potrebám.

Jednou z možností je radenie zadávaných úloh od zákazníka podľa priorít. Ak ich je príliš veľa, tak tie najmenej potrebné sa do projektu vôbec nedostanú alebo sa použijú až v ďalšej verzii produktu. Tento krok je však potrebné konzultovať s klientom. Taktiež treba dohliadnuť na konzistenciu úloh, ktoré sa budú implementovať, aby sa nestalo, že sa implementuje nejaká vlastnosť ktorá je dôležitá pre inú, čo sa do projektu nedostala.

Ďalším spôsobom vylepšenia je nepoužívať len jeden, ale dva plány [1]. Jeden sa prezentuje zákazníkovi a dohodujú sa na ňom termíny. Druhý plán je interný a pracuje s ním tím. Interný plán obsahuje oproti verejnému rezervu, z ktorej je možné čerpať ak by sa vyskytli nejaké problémy. Vo všeobecnosti existujú dve možnosti, ako rezervu získať. Prvou je jej ukrojenie z celkového vypočítaného času trvania všetkých úloh. Táto možnosť však nie je veľmi výhodná, lebo dopredu sa spoliehame na to, že čas implementácie bude kratší ako je v skutočnosti možný a mohlo by prichádzať k vyťažovaniu pracovníkov. Druhou je jej pridanie k celkovému času. Je však potrebné si dávať pozor, aby sa o ňom nedozvedel zákazník lebo by nemusel byť veľmi nadšený, že v skutočnosti platí aj za čas navyše. Taktiež vzniká aj nebezpečenstvo nezískania zákazky, pokiaľ konkurencia prezentuje kratší termín dodania.

Kontrola plánu

Jednou z najdôležitejších súčastí plánovania je jeho kontrola. Len pomocou nej vieme zistiť či postupujeme správne a či sa nám podarí dodržať stanovené termíny. Aké spôsoby kontroly môžeme použiť pri našom spôsobe plánovania?

Pri opisovanom kombinovanom spôsobe plánovania existujú dva spôsoby kontroly. Jednou je kontrola dodržovania pevne stanovených termínov založená na iteratívnom plánovaní a druhou je porovnávanie stráveného času na úlohách s odhadom založeného na evidencii. Z pozorovania strávených časov na úlohách už vieme presne zistiť, či sa nám podarí dodržať stanovené termíny a podľa priebežných termínov aj finálny termín projektu. Vieme tak celkom pružne reagovať a v prípade problémov je možné projekt preplánovať.

Ako zhodnotiť správnosť plánu?

Ako môžeme rozpoznať, že náš projekt je nesprávne naplánovaný? Jednou z možností je porovnať náš plán s deviatimi omylmi plánovania podľa Steve McConnella, ktoré sú [2]: neplánovanie, neplánovanie dostatočne, neplánovanie rizík, používanie rovnakého plánu pre všetky projekty, používanie cudzích plánov, odklánanie plánu od reality projektu, plánovanie do veľkých detailov priskoro, plánovanie s krátkym časom trvania a neučenie sa z predchádzajúcich problémov.

Prvým omylom je neplánovanie projektu. Keďže plánovať sme začali, tak sa nás tento bod príliš netýka, úspešne sme sa mu vyhli a nie je pre nás viac podstatný.

Ďalším bodom je nedostatočné plánovanie. Pretože náš plán závisí od presných termínov a od dostatočného naplánovania jednotlivých úloh, tak táto podmienka je

dodršaná. Inak by sa zvyšovalo riziko neurčitosti a teda by sme nevedeli povedať, či sa nám podarí stanovený termín dodržať.

Ako zdroj možných problémov pri plánovaní softvérových projektov môže byť aj situácia, keď sa objaví nečakaná udalosť, ktorá nebola odhalená pri analyzovaní potenciálnych rizík. Takto vzniknutá situácia, s ktorou nebolo pri zostavovaní časového plánu kalkulované, spôsobuje, že hoci bol pôvodný plán dobre navrhnutý a bola vytvorená aj predpokladaná časová rezerva vzhľadom na možné riziká, objavením sa väčšieho počtu nepredpokladaných rizík sa môže stať časová rezerva nedostatočnou. Plánovanie založené na evidencii využíva iný prístup pri kalkulácii s rizikami, keďže riziká ako také neberie do úvahy. Vychádza totiž už zo štatistických údajoch o predchádzajúcich projektoch, ktoré zahŕňajú popri čase nevyhnutnom na samotnú implementáciu aj čas kedy sa na daných projektoch odstraňovali chyby alebo vzniknuté rizikové udalosti.

Nie všetky projekty, ale aj úlohy na projekte sú rovnakého typu. Môžu sa odlišovať rôznymi spôsobmi implementácie, ktoré závisia od presných požiadaviek zákazníka. Preto je potrebné zohľadniť aj predchádzajúcu evidenciu na rôzne typy úloh a pri odhadovaní času použiť tie správne.

Pretože plánovanie je závislé aj od správnych odhadov a tie od predchádzajúcej evidencie jednotlivých vývojárov, je potrebné ju aj použiť. Len podľa nej vieme odhadnúť čo najpresnejšie trvanie úlohy. Pokiaľ by sme použili cudzie štatistiky, tak opäť strácame istotu dodržania termínov. Tento problém nastáva pri prvotnom plánovaní pomocou evidencie alebo pokiaľ na projekt príde nový vývojár. Stále je však lepšie použiť pri odhade generické relatívne rýchlosti vykonávania ako nepoužiť žiadne. Túto odchýlku však musíme pri projekte zohľadniť.

Dôležitým prvkom na porovnanie plánu s aktuálnym stavom projektu je jeho neustála kontrola. Je preto potrebné pôvodný plán aj neskôr používať a v prípade problémov úlohy opätovne preplánovať. Nemenej dôležité je aj zbieranie evidencie za jednotlivé úlohy a vývojárov, aby sme v budúcnosti vytvorili ešte dokonalejšie plány.

Najkontroverzejším omylom plánovania je plánovanie softvérového projektu do veľkých detailov priskoro. Jednak potrebujeme čo najlepšie odhadnúť celkový čas trvania projektu, ale zároveň sme aj časovo obmedzení pridávaním novej funkcionality do nasledujúcej implementácie projektu. Termín zároveň slúži aj ako ochrana pred príliš dlhým plánovaním projektu.

Pokiaľ máme nazbieranú dostatočnú evidenciu z minulosti, tak vyhnúť sa naplánovaniu projektu či úloh s odhadovaným príliš krátkym časom trvania by nemal byť problém. V kombinácii s dostatočnou časovou rezervou, z ktorej je možné čerpať pri omeškaní na konci projektu sa plán ešte viac zdokonalí.

Pri plánovaní nového projektu je dôležité si zobrať ponaučenie aj z predchádzajúcich. Pokiaľ sa relatívne rýchlosti na jednotlivé úlohy veľmi odlišujú, je potrebné navýšiť množstvo prideleného času na vývoj. Rovnako to je aj pri testovaní. Pokiaľ sa v minulosti nachádzalo v úlohách množstvo chýb, tak je predpoklad, že sa budú nachádzať aj v ďalšom projekte. Takýchto príkladov existuje mnoho a je potrebné ich zohľadniť aj pri novom projekte, pretože inak by sme vystavovali ďalšiemu riziku.

Záver

Kombinácia rôznych prístupov plánovania softvérového systému môže viesť ešte k lepším výsledkom ako sú dosahované samostatne. Pri ich kombinovaní sa však môže dostaviť aj opačný účinok, ktorý nie je želaným, preto je dôležité vybrať správne kombinácie.

Esej zachytáva spôsob plánovania, ktorý sa snaží čo najviac minimalizovať neurčitost' a čo najviac zrealizovať celkový rozvrh projektu a zároveň vychádza z kombinácie výhod rôznych riešení. Keďže je dôležité dodržať finálny termín ukončenia, navrhnuté riešenie obsahuje pevne stanovené termíny, dokedy je možné navrhovať novú funkcionálnosť či projekt implementovať.

Vo všeobecnosti aj tak nezáleží, aký spôsob plánovania sa nakoniec vyberie. Dôležité je aby sa vôbec plánovalo, lebo len pomocou plánu vieme znížiť neistotu a spoznať väčšinu nástrah, ktoré nás pri projekte čakajú. Je to nástroj, ktorý nám ukáže cestu do cieľa aj keď celú ju spoznáme až v jej závere. Bez plánu by bol cieľ len snom, ktorý sa nemusí naplniť.

Použitá literatúra

1. Armour, Phillip G.: To plan, two plans, *Communications of the ACM*, Volume 48 Issue 9, Pages 15 - 19 , 2005.
2. McConell, Steve: The Nine Deadly Sins of Project Planning, *IEEE Software*, Vol. 18, No. 5, 2001.
3. O'Connell, Fergus: *How to Run Successful Projects II*, the silver bullet, Prentice Hall, Englewood Cliffs, NJ., 1996.
4. Rothman, Johanna: *Iterative software project planning and tracking*, 1997 <http://www.jrothman.com/1997/01/iterative-software-project-planning-and-tracking/> [cit. 2012-10-06].
5. Spolsky, Joel: *Evidence Based Scheduling*, 2007 <http://www.joelonsoftware.com/items/2007/10/26.html> [cit. 2012-10-06].

Annotation

Exploitation the potential of different approaches in project planning

Planning is one of the most important stages of project management related to the creation of a software product. Over time, different scheduling techniques by which we can successfully complete the project in the required time and required quality are rising. Each method has its own advantages and disadvantages, so it is necessary to choose the right one, or customize existing one for the own needs or the needs of the project. Essay presents different scheduling options. Intends to why and when they would be appropriate or inappropriate to use and suggests a better solution that minimizes their deficiencies and increase the realism of created schedules. The suggested solution also contains mechanisms to avoid constantly changing specifications and the associated prolongation of the product implementation time. Essay thinking of the possible extensions of the suggested solution by which it is possible to have even better results, but also they may carry some risks.