

Kol'ko to bude stát'?

Odhady v softvérových projektoch

TOMÁŠ MATUŠEK

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
matusek01@student.fiit.stuba.sk*

Abstrakt. V tvrdej konkurencii vždy zvíťazí ten, kto dokáže najlepšie odhadnúť situáciu a pružne reagovať na podnety z okolia. Zvlášť to platí v oblasti vývoja softvérových produktov. Príspevok stručne definuje základné fázy procesu odhadovania a poukazuje na dôležitosť využitia skúseností z predchádzajúcich projektov pri jeho realizácii. Ďalej sa pokúša rozoznať najdôležitejšie problémy, ktorým musia tvorcovia odhadov čeliť a ponúka k nim možné riešenia. Nakoniec uvádza cenné rady a pripomienky, ktoré majú prispieť k lepšiemu odhadu, a teda aj k úspešnejšiemu projektu.

Úvod

Odhad a naplánovanie času a úsilia potrebného k vývoju softvérového systému patria medzi úlohy s vôbec najvyšším sklonom k chybovosti v oblasti softvérového inžinierstva. Dôvodov, prečo tomu tak je, existuje množstvo, z nich najdôležitejšie sú [1]:

- Nekvalifikovaný odhad: nemôžeme dostatočne odhadnúť nič, čomu nerozumieme.
- Neuvažovanie prostredia: všetky techniky odhadu je potrebné upraviť podľa podmienok prostredia, v ktorom ich chceme aplikovať.
- Nejasný proces odhadu: musia byť stanovené a dodržiavané presné procedúry odhadu.
- Neschopnosť realizácie odhadu: podmienky stanovené odhadom nesmú byť porušené.
- Nedostatočná flexibilita odhadu: je potrebné si uvedomiť, že softvérový projekt nie je statická entita, a teda sa vyvíja v čase. Rovnako by sa mal meniť aj náš odhad.

Je všeobecne známe, že teória a prax majú od seba na míle ďaleko. Existuje množstvo prístupov alebo metód odhadovania, ktoré sú podrobne vysvetlené v rôznych odborných prácach a pomerne ľahko dostupné. Cieľom môjho príspevku je skôr upozorniť na problémy, ktoré sa môžu objaviť v rôznych štádiách odhadovania, chyby, ktorým sa je potrebné vyhnúť a ponúknuť nápady a vylepšenia, ktoré prispievajú k lepšiemu odhadu a tým aj k úspešnému projektu. Verím, že úsilie mnou vynaložené nebolo zbytočné a prinesie svoje ovocie.

Prehľad činností

Proces odhadovania softvérových projektov pozostáva z týchto štyroch základných krokov [2]:

1. Odhad rozsahu vyvíjaného produktu.
2. Odhad úsilia (v človeko-mesiacoch alebo človeko-hodinách).
3. Plán činností (v kalendárnych mesiacoch).
4. Odhad ceny projektu (v peniazoch).

Každý z krokov si stručne vysvetlíme, aby sme získali prehľad a lepšie dokázali pochopiť a identifikovať možné problémy.

Odhad rozsahu

Presný odhad rozsahu projektu je prvým krokom k úspechu. Zdroj informácií by mal, pokiaľ je to možné, vychádzať z formálnej špecifikácie požiadaviek. V praxi je však obvyklé, že jediné, čo máme, je slovný opis. Napriek tomu sa nesmieme nechať odradiť a riziká vyplývajúce z nepresného opisu zakomponovať do odhadu a oboznámiť s nimi účastníkov projektu. Odhad je treba prehodnotiť okamžite, ako sú k dispozícii presnejšie údaje.

Odhad sa vykonáva dvoma základnými spôsobmi:

1. Analogicky, t.j. vychádza sa z predchádzajúceho projektu, ktorý mal podobné vlastnosti. Projekt sa rozdelí na niekoľko častí, ktorým sa prideli percentuálne ohodnotenie voči predchádzajúcemu projektu. Celkovo sa dá dospieť k pomerne presným výsledkom za predpokladu, že projekty sú si dostatočne podobné.
2. Použitím algoritmických postupov, ako napr. metóda funkčných bodov (Function points) [2]. Počítajú a vyhodnocujú sa vlastnosti produktu.

Odhad úsilia

Podarilo sa nám vytvoriť a identifikovať predpokladaný rozsah projektu, takže z neho teraz môžeme odvodiť potrebné úsilie, ktoré treba vynaložiť. Táto konverzia je možná len vtedy, ak máme definovaný vývojový cyklus softvéru a procesy s ním súvisiace. V súvislosti s výberom procesov a vhodného vývojového cyklu treba spomenúť fakt, že menšie projekty sa riadia ľahšie ako veľké [3]. Teda je vhodné si projekt postupne

rozdeliť na niekoľko samostatných modulov a venovať sa odhadom pre každý z nich samostatne. Výhody vyplývajúce z takéhoto rozhodnutia sú:

- Plány a odhady sú presnejšie.
- Chyby a nedostatky sa hľadajú rýchlejšie.
- Pokrok vo vývoji sa dá jasne sledovať.
- Pracovná morálka je vyššia (každý má pocit, že prispel k pokroku).

Samotný odhad úsilia si vyžaduje identifikovať a kvantifikovať všetky aktivity potrebné na vytvorenie produktu odhadovanej veľkosti. Na výber máme z dvoch alternatív:

1. Na základe záznamov o priebehu predchádzajúceho projektu môžeme jednoducho stanoviť potrebné úsilie pre aktuálny projekt. Avšak len za predpokladu, že záznamy existujú, jedná sa o projekt porovnateľného rozsahu, využívame rovnaký vývojový cyklus a na vývoji sa budú podieľať tí istí ľudia.
2. Ak z nejakého dôvodu nemôžeme použiť prvý spôsob, riešením sú všeobecne akceptované algoritmické prístupy ako COCOMO alebo Putnam metodológia [2], ktoré vychádzajú zo štúdia množstva úspešne skončených projektov.

Plán činností

Vytvorenie plánu v sebe vo všeobecnosti zahŕňa počet ľudí, ktorí budú pracovať na projekte, náplň ich činností a čas, kedy prácu na projekte začnú a kedy ju ukončia. Keď máme tieto informácie, je potrebné ich premietnuť do kalendárneho rozpisu. Znovu je najlepšie využiť skúsenosti nadobudnuté z predchádzajúcich projektov.

Odhad ceny

Pri stanovovaní celkovej ceny projektu existuje veľa faktorov, ktoré musíme brať do úvahy. Je potrebné uvažovať náklady na mzdy, hardvér a softvér, priestory, cestovné, školiace kurzy a podobne. Presné stanovenie ceny závisí od finančnej politiky spoločnosti. Najjednoduchšie stanovenie ceny práce je vynásobenie potrebného úsilia hodinovou sadzbou. Samozrejme, takto jednoducho to v praxi nejde, keďže každá úloha v tíme je spravidla inak finančne ohodnotená. Znovu, najlepší spôsob je vychádzať zo skúseností z predchádzajúcich projektov.

Presnosť odhadu a kompromisy

Vždy, keď sa vytvorí odhad si kladieme otázku, do akej miery zodpovedá skutočnosti. Presná odpoveď neexistuje, aspoň nie do ukončenia projektu. Pochopiteľne, každý si želá, aby bol odhad vychádzajúci z dostupných údajov v každej fáze čo najpresnejší. Dôležité je, aby odhad nevytváral falošný dojem istoty.

Čo to znamená? Výsledok odhadu by mal tvoriť nejaký interval hodnôt, ktoré ohraničujú danú oblasť. Čím skúsenejší je tvorca odhadu, tým je interval užší. Nie je správne používať presné čísla, pretože zvädzajú k domnienke, že aj odhad je presný, čo samozrejme nie je pravda, pretože v praxi je veľmi málo vecí jednoznačných a presných. Presnosť odhadu ďalej ovplyvňujú nasledovné skutočnosti:

- Presnosť údajov používaných pri odhade.
- Presnosť výpočtov pri odhade.
- Miera, s akou sa zhodujú využívané záznamy z predošlých projektov s aktuálnym projektom.
- Predvídateľnosť vývojového procesu.
- Neočakávané okolnosti.

Po vytvorení odhadu sa môžeme pustiť do riešenia ďalšieho problému – ako skombinovať zistené skutočnosti, aby sme uspokojili seba, manažment a zákazníkov. Táto úloha si vyžaduje dobrú znalosť jednotlivých faktorov a vzťahov medzi nimi. Je dôležité mať na pamäti tieto veci:

- Cena produktu a čas vývoja sú nepriamo úmerné. Predĺžením času vývoja je možné ušetriť, riskujeme však nespokojnosť zákazníka. Je potrebné zvoliť vhodný kompromis, t.j. zistiť, aké omeškanie je pre zákazníka ešte prípustné.
- Urýchliť vývoj sa dá len tromi spôsobmi. Znížením funkcionality systému, zvýšením počtu pracovníkov (ale len pri činnostiach, ktoré je možné vykonávať paralelne) alebo predĺžením pracovnej doby. Funkcionalita by nám neprešla u zákazníkov, takže do úvahy prichádzajú len druhé dve možnosti, čo má však za následok nárast ceny. Navyše, väčší počet ľudí nemusí znamenať viac času. Pri predĺžení pracovnej doby síce produktivita dočasne stúpne, ale po čase klesne, keďže ľudia budú viac vyčerpaní.
- Pre každý projekt existuje tzv. minimálny možný časový plán, ktorý sa treba snažiť nájsť.

Je zaujímavé pozorovať dôsledky niektorých kompromisov na konečnú cenu produktu. Ako príklad je možné uviesť projekt, kde rozdiel medzi najkratším a nominálnym časovým plánom je iba 2 mesiace., ale na to treba zvýšiť počet ľudí o 10 a celkové náklady vzrastú skoro o 870.000\$ [2].

Nie v každom projekte je takýto drastický rozdiel medzi najkratším a nominálnym časovým plánom, ale vzťah medzi cenou a počtom zamestnancov podlieha pravidlám, ktoré nie je možné ovplyvniť, a preto je potrebné, aby s nimi každý zúčastnený počítal.

Problémy

Hoci je efektívny odhad jednou z najdôležitejších častí tvorby softvérového produktu, sú s ním najväčšie problémy. Prečo je to také ťažké? Nasledujúce dôvody nám na túto otázku odpovedia [2].

- Odhadnúť rozsah projektu predstavuje veľmi ťažkú úlohu, ktorá sa zvyčajne opomína a prechádza sa priamo k tvorbe rozvrhu. Samozrejme, je to nesprávny prístup, pretože keď sa nezamyslíme nad tým, čo máme za úlohu vytvoriť, asi sa nám nepodarí vymyslieť dobrý rozvrh a tým pádom úspešne skončiť projekt.
- Zákazníci si zvyčajne neuvedomujú, že vývoj softvéru je proces neustáleho vylepšovania a zmeny a odhady vytvorené na začiatku procesu sú pomerne nepresné. Aj dobré odhady sú iba odhady so svojou dávkou neurčitosti, a predsa sú často pokladané za istotu. Riešením problému by mohlo byť vytváranie odhadov ako intervalov určitej šírky namiesto konkrétnych dátumov. Je treba si dať pozor, aby interval nebol príliš úzky, lebo tým sa nič nerieši. Ďalším možným východiskom je vyjadrovať sa o termínoch s určitou pravdepodobnosťou, napr. je na 80 % isté, že to stihnem do určitého dátumu.
- Spoločnosti zvyčajne nezaznamenávajú údaje o vyvíjaných projektoch, pritom práve tieto významne prispievajú k úspešným odhadom. Je preto potrebné, aby sa zaviedli nejaké metriky, na základe ktorých sa môžu údaje vyhodnocovať.
- Často je ťažké dohodnúť sa na reálnom termíne so zákazníkom alebo manažmentom. Každý chce mať všetko hotové čo najskôr, avšak existuje minimálna možná doba, pod ktorú sa čas vývoja projektu nedá stlačiť. Treba dobre zvážiť, čo je a nie je možné a za akú cenu a neustúpiť pod tlakom zákazníka. Záznaky sa síce stávajú, ale nie tak často, aby sa na ne dalo spoliehať.

Špecifické prípady

Malé projekty

Veľa ľudí pracuje na malých projektoch, ktoré sú definované malým počtom účastníkov (jeden až dvaja ľudia) a krátkym časovým rozsahom (do 6 mesiacov) [2]. Algoritmické postupy založené na veľkých projektoch sa na ne dajú veľmi ťažko aplikovať. Odhady v malých projektoch sú vysoko závislé na vlastnostiach individuálnych riešiteľov, a preto je najlepšie, aby ich vykonávali sami.

Projekty v neznámom prostredí

Čoraz častejšie sa stáva, že riešime projekt z oblasti, v ktorej nikto z tímu nemá žiadne skúsenosti. Takéto projekty so sebou nesú vysokú mieru rizík a je potrebné postupovať mimoriadne opatrne. Riziká treba dôkladne identifikovať a oboznámiť s nimi zákazníkov. Rovnako je nevyhnutné sa vyhnúť záväzkom v podobe pevných termínov dokončenia. Po každom zlepšení orientácie v problémovom prostredí je vhodné prehodnotiť odhad.

Výber vývojového cyklu projektu je často kľúčový krok pre úspech softvéru. Pri projektoch s vysokou mierou neurčitosti sú vhodné najmä iteratívny a špirálový model, keďže vývoj sa vykonáva po kúskoch a v každej fáze dochádza k identifikácii rizík a je možné prehodnotiť odhad. Naopak, klasický vodopádový model je z jasných príčin nevhodný.

Tipy a vylepšenia

Nasledujúce zásady majú za úlohu vylepšiť proces odhadovania v softvérových projektoch.

- Na realizáciu odhadu si treba rezervovať dostatok času. Neuvážené odhady vedú k zvýšeniu rizík a v konečnom dôsledku k zlyhaniu projektu.
- Všade, kde je to možné, sa využívajú údaje získané z prechádzajúcich projektov.
- Odhady by mali vykonávať ľudia, ktorí v danej oblasti pracujú. Inak sa zvyšuje nepresnosť.
- Nástroje na tvorbu odhadov predstavujú veľkú pomoc. Uľahčujú samotný odhad a starajú sa o to, aby sa na nič nezabudlo.
- Každý odhad vykonáva viacero ľudí a používajú sa rôzne metódy. Porovnaním výsledkov sa dospeje buď k ich konvergencii, t.j. odhad je pravdepodobne dobrý alebo k rozptylu, čiže asi sme niečo prehliadli a je potrebné lepšie pochopiť skúmaný problém.
- Počas životného cyklu projektu je vhodné vykonávať odhady pravidelne. Spolu s produktom sa vyvíjajú aj odhady, ktoré sa postupne blížia k skutočnej hodnote.
- Štandardizovaný proces odhadu, ktorý každý chápe a dodržiava, znamená lepší prehľad a efektívnejšie rozdelenie úsilia.
- Časť úsilia je potrebné venovať aj vylepšovaniu procesu odhadu. Po každom ukončenom projekte sa vyhodnotia výsledky, zistia sa chyby a nedostatky a vykoná sa prípadná náprava.

Zhrnutie

Ľudia sa odjakživa snažili nájsť univerzálne riešenie na všetky problémy. Samozrejme, nikdy neuspeli, pretože svet nie je čiernobiely ani statický, mení sa z minúty na minútu a nie je možné eliminovať neurčitost', ktorá nás obklopuje.

Neexistuje dokonalý odhad ani žiadna metóda, ktorá by nám ho priblížila. Žiaden algoritmus nikdy nezaručí úspech, nech je akokoľvek skvelý. Preto jediné, čo nám ostáva, je učiť sa a získavať cenné skúsenosti.

Človek sa síce najlepšie učí na vlastných chybách, ale tie so sebou nesú skoro vždy negatívne dôsledky. Kto sa im chce vyhnúť, používa metódu možno menej účinnú, zato bezpečnejšiu. Učí sa na chybách a skúsenostiach druhých. A presne o to som sa pokúsil v mojom príspevku. Priblížiť problematiku odhadovania a upozorniť na chyby a skúsenosti iných. Dúfam, že sa mi to podarilo.

Použitá literatúra

1. Fairley, R.E.: Recent advances in software estimation techniques. *Proceedings of the 14th international conference on Software engineering*. ACM Press, 1992.
2. Peters, K.: Software project estimation, Software Productivity Center, 1999.
3. Rettig, M, Simons, G.: A project planning and development process for small teams. *Communications of the ACM*, Vol. 36, No. 10 (1993).

Annotation

How much?

In today's ruthless competition the winner is always the one who makes the best estimate of the situation and reacts accordingly. This is also true for software engineering. This paper briefly defines basic phases of estimation process denoting the importance of reusing historical data from similar projects. It also tries to identify serious problems, which the estimators must face and proposes alternative solutions. Finally, it presents valuable tips and suggestions, which contribute to better estimates and eventually, successful project.