

# PRESNÝ ODHAD? NEMOŽNÉ!

*Iba hlupák verí predpovedi budúcnosti.*

*Andrej Belica*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
belicadud0 [zavináč] gmail [.] com

**Abstrakt.** *Odhad softvérového projektu môže veľmi ovplyvniť proces jeho plánovania, jeho priebeh, aj jeho výsledok. Ako ich ovplyvní, závisí od jeho presnosti. Všeobecne platí, že čím presnejší odhad, tým lepšie. Ale ako dosiahnuť presný odhad? Je to vôbec možné? Vytvorenie dobrého odhadu softvérového projektu je nákladné a náročné. Oplatí sa pre všetky projekty vytvárať odhady? Ak áno, ako by sa mali vytvárať? Na tieto, ale aj iné otázky hľadám odpoveď v mojej eseji. Venujem sa niekoľkým metódam odhadovania softvérových projektov, ich náročnosti a presnosti. Zamýšľam sa nad potrebou odhadovania rôznych typov softvérových projektov a nad tým, ako by sa mali v týchto projektoch vytvárať odhady. V neposlednom rade rozoberám aj problematiku odhadovania vzhľadom na náš tímový projekt.*

**Kľúčové slová:** *Odhad, softvérový projekt.*

## Úvod

V súčasnej dobe komplexného a zložitého softvéru sa žiadny väčší softvérový projekt nezaobíde bez plánovania. Na tom sa zhodne asi väčšina projektových manažérov a aj to praktikujujú vo svojich projektoch. Ale aj tak je veľká časť projektov neúspešná (nie sú dokončené načas alebo majú príliš veľké náklady). Na vytvorenie dobrého plánu je potrebné vytvoriť množstvo odhadov. Treba odhadnúť rozsah projektu, potrebné množstvo vynaloženého úsilia, náklady, čas trvania jednotlivých úloh atď. Keď hovoríme o rozsahu projektu, tak nám ide hlavne o množstvo úsilia, ktoré treba vynaložiť na dokončenie projektu. Na základe toho treba určiť, aký veľký tím bude treba a ako dlho bude projekt trvať. Z týchto a ďalších parametrov potom vyplývajú náklady projektu.

Nepresnosť odhadov je jednou z hlavných príčin neúspešnosti softvérových projektov. Preto by bolo ideálne, aby naše odhady boli čo najpresnejšie. Nemali by byť

hádáním, ale mali by byť založené na faktoch. Musíme čo najlepšie využiť všetky fakty, ktoré sú nám o projekte známe a na ich základe odhadovať to, čo ešte o ňom potrebujeme vedieť.

Jedným z hlavných problémov pri odhadovaní býva nedostatok odhadových skúseností projektového manažéra alebo iného člena tímu, ktorý vytvára odhady. Ďalším problémom býva nedostatok údajov, na ktorých sa dajú odhady zakladať. Môžu to byť napríklad údaje z predchádzajúcich projektov, ktoré sú relevantné pre náš aktuálny projekt. Podstatným problémom tiež môže byť nepoužitie systematického procesu odhadovania. Toto sa podarilo potvrdiť v práci [3].

## Predpovede budúcnosti

Všeobecne platí, že keď plánujeme, musíme vedieť povedať čo sa stane, predtým ako sa to stane. To znamená, že potrebujeme predpovedať budúcnosť. O takéto predpovede sa snažia ľudia už od dávnych čias. V minulosti sa veštcí a proroci snažili predpovedať budúcnosť s pomocou bohov. Dnes na to projektoví manažéri používajú rôzne sofistikované metódy a pomoc expertov. Oni už vedia, že budúcnosť sa nedá predpovedať, preto ju iba odhadujú.

V softvérových projektoch je dôležité v prvom rade odhadnúť aký rozsiahly bude projekt. Rozsah projektu sa meria viacerými spôsobmi. Často používaným meradlom je počet riadkov zdrojového kódu (angl. Source Lines of Code – SLOC). Táto metóda naznačuje, že čím viac riadkov bude zdrojový kód programu obsahovať, tým viac programátorov bude potreba alebo tým dlhšie bude jeho vývoj trvať, alebo čím viac riadkov kódu je napísaných, tým bližšie je projekt k dokončeniu. Je do určitej miery závislá od použitého programovacieho jazyka. Nemyslím si, že tento spôsob veľmi dobre odráža množstvo funkčnosti, ktorá bude do projektu implementovaná. Je to ako tvrdiť o knihe, že čím má viac strán, tým viac vedomostí obsahuje.

Ako lepšie meradlo rozsahu projektu sa mi preto javia funkčné body (angl. Function Points – FP). Ako meradlo rozsahu projektu sa používajú obrazovky, správy, tabuľky, dotazy alebo rozhrania. Tento spôsob realistickejšie reprezentuje úsilie, ktoré je potrebné vynaložiť na dokončenie projektu. Ale nemyslím si, že tento model je univerzálne lepší pre všetky projekty. Pre projekty, ktoré sú skôr výpočtovo a algoritmicky zamerané sa častejšie používa počet riadkov zdrojového kódu.

Ani jeden z týchto spôsobov nie je ideálny, pretože sa nedá presne odvodiť, koľko úsilia je treba vynaložiť na základe počtu riadkov zdrojového kódu alebo funkčných bodov. Napriek tomu sa oplatí odhadnúť rozsah projektu pomocou týchto meradiel, pretože je to jednoduchšie a stále presnejšie ako odhadovať priamo vynaložené úsilie.

V [2] autor tvrdí, že ani jedno z týchto meradiel nie je úplne vhodné na určovanie rozsahu projektu. Navrhuje, aby sa bola vynájdená určitá „jednotka znalosti“, pomocou ktorej by sa dalo presne povedať, koľko práce treba do projektu vložiť, aby bol dokončený. S touto myšlienkou súhlasím, ale nemyslím si, že v blízkej budúcnosti by mohlo niečo také byť vymyslené.

Čo sa týka samotného odhadovania, postupy sa rozdeľujú na dve hlavné skupiny. *Parametrickými* (tiež sa môžu nazývať algoritmické) metódami sa rozsah odhaduje pomocou aritmetických modelov na základe koeficientov a vonkajších veličín. Výsledky

môžu byť nepresné, pretože modely bývajú väčšinou jednoduché a preto neberú do úvahy všetky dôležité faktory. Tento problém je riešiteľný a je aj predmetom výskumu ale napriek tomu sa parametrické prístupy veľmi v praxi nepoužívajú.

Aj keď sa vyvíja množstvo úsilia na výskum vhodných aritmetických modelov pre parametrické metódy, v praxi je najviac používaná druhá skupina metód, ktorou sú *heuristické* prístupy. Pri týchto metódach sa nepoužívajú vonkajšie parametre, ale úsudok expertov. Je viacero možností, ako pristupovať k odhadom.

Pri metóde *zdola nahor* (angl. bottom up) sa na projekt pozeráme ako na množinu modulov. Pre každý modul odhadujeme úsilie zvlášť. Celkové úsilie potrebné na projekt sa potom určí sumou úsilia potrebných na vytvorenie jednotlivých modulov. Pri tejto metóde by bolo podľa mňa vhodné, aby odhady nevytváral iba projektový manažér, ale úsilie pre každý modul by mal odhadnúť človek zodpovedný za jeho tvorbu. Projektový manažér by potom tieto odhady prehodnotil a následne akceptoval, prípadne vrátil na opakovaný odhad.

Nevýhodou tejto metódy je, že sa nehľadí na úsilie potrebné na integráciu modulov do finálneho produktu a na iné činnosti.

Inou možnosťou pri tomto prístupe je nerozoberať projekt na jednotlivé moduly, ale na činnosti, ktoré treba vykonať na dokončenie projektu. Výsledné úsilie je potom súhrnom úsilia potrebného na vykonanie jednotlivých činností.

Pri metóde *zhora nadol* (angl. top down) odhadujeme úsilie pre projekt ako celok a potom ho rozdeľujeme pre jednotlivé časti projektu. Vychádzame pritom zo všeobecných vlastností projektu. Často sa používa *analógia*, to znamená, že si pomáhame skúsenosťami z predchádzajúcich podobných projektov. Táto metóda tým pádom zahŕňa všetko úsilie spojené s projektom, ale nezaobera sa implementačnými detailmi. Preto pri takomto odhade riskujeme možnosť, keď napriek tomu, že sa projekty podobajú, súčasný projekt bude obsahovať nejaké časti, ktoré budú vyžadovať podstatne viac alebo menej úsilia ako pri predchádzajúcich projektoch. Tiež môže nastať nejaká nečakaná udalosť, s ktorou sa odhadujúci v predchádzajúcich skúsenostiach nestretol.

Metóda odhadu *Delphi* je ďalšou metódou založenou na úsudkoch expertov. Pri tejto metóde niekoľko (mali by byť minimálne štyria) experti spolupracujú na vytvorení odhadu. Každý z nich anonymne vyplní formulár odhadu. Potom dostanú tabuľku so všetkými odhadmi, pričom nevedia, komu patrí ktorý odhad. Na základe výsledku sa môžu dohodnúť či výsledný odhad akceptujú alebo budú proces opakovať. Medzi jednotlivými odhadmi môžu problém diskutovať.

Táto metóda je objektívnejšia ako ostatné, pretože nie je výsledkom práce jedného experta. Výsledné odhady môžu byť veľmi presné. Na druhej strane je táto metóda aj nákladnejšia ako ostatné. Napriek tomu je často používaná pri odhadovaní projektov, hlavne u väčších spoločností.

O žiadnej z týchto metód sa nedá jednoznačne povedať, že je najvhodnejšia pre všetky projekty a vždy poskytne kvalitné odhady. Spomenuté metódy majú svoje výhody a nevýhody, ako aj obmedzenia. A pri presnosti odhadu vždy záleží hlavne na ľuďoch, ktorí ho vykonávajú. Všetky metódy založené na úsudku človeka (experta) sú subjektívne.

V tabuľke (Tab. 1.) sú stručne vyjadrené výhody a nevýhody spomínaných prístupov odhadovania. Tabuľka je založená na tabuľke z [1] a je mierne upravená.

Tab. 1. Výhody a nevýhody metód odhadovania

Metóda	Výhody	Nevýhody
Algoritmické metódy	<ul style="list-style-type: none"> <li>• Objektívne, opakovateľné, analyzovateľné</li> <li>• Objektivita kalibrovaná na skúsenosti</li> </ul>	<ul style="list-style-type: none"> <li>• Subjektívne vstupy</li> <li>• Hodnotenie výnimočných okolností</li> <li>• Kalibrované na minulosť</li> </ul>
Analógia	<ul style="list-style-type: none"> <li>• Založená na reprezentatívnej skúsenosti</li> </ul>	<ul style="list-style-type: none"> <li>• Reprezentatívnosť skúsenosti</li> </ul>
Zhora nadol	<ul style="list-style-type: none"> <li>• Sústredenie na úrovni systému</li> <li>• Efektívna</li> </ul>	<ul style="list-style-type: none"> <li>• Menej detailná</li> <li>• Menej stabilná</li> </ul>
Zdola nahor	<ul style="list-style-type: none"> <li>• Detailnejšia</li> <li>• Stabilnejšia</li> <li>• Podporuje zaviazanie sa jednotlivcov</li> </ul>	<ul style="list-style-type: none"> <li>• Môže prehliadnúť náklady na úrovni systému</li> <li>• Vyžaduje viac úsilia</li> </ul>
Delphi	<ul style="list-style-type: none"> <li>• Objektívnejšia</li> </ul>	<ul style="list-style-type: none"> <li>• Vyžaduje viac úsilia</li> </ul>

V ďalšej časti sa budem venovať rôznym aspektom používania týchto praktík pri odhadovaní softvérových projektov.

### Kedy, kde a ako odhadovať

Nech sa akokoľvek snažíme, presný odhad dosiahneme, iba ak máme šťastie. Žiadna metóda nám nezaručí presnú predpoveď. Je to tak aj preto, lebo v čase odhadovania, jednoducho nemáme dostatok údajov k tomu, aby sme predpovedali to, čo potrebujeme. Keby sme tie údaje mali, tak nemusíme odhadovať, ale stačí jednoducho odvodiť výsledok. A čím menej údajov máme, tým ťažší a nepresnejší bude náš odhad.

Logicky by malo platiť, že čím skôr v priebehu projektu robíme odhad, tým menej máme údajov o projekte. Preto by sme mali odhady robiť nielen na začiatku projektu, ale aj neskôr v jeho priebehu, keď máme o ňom viac údajov.

Počas procesu odhadovania nemusíme používať iba jednu metódu odhadovania. Pre realistický odhad je vhodné ich použiť viac. Minimálne si myslím, že sa vždy oplatí použiť analógiu a využiť poznatky z predchádzajúcich projektov, ak takéto poznatky máme k dispozícii.

Vhodnosť použitia rôznych metód odhadovania závisí do veľkej miery aj od veľkosti a typu projektu a od veľkosti a skúseností tímu a spoločnosti, ktorá projekt vytvára.

Najlepšie na tom je veľká spoločnosť, ktorá má za sebou množstvo projektov. Skúsenosti z týchto projektov jej nezanedbateľne uľahčia proces odhadovania nového projektu. Pri veľkej spoločnosti môžeme počítať s tým, že disponuje veľkým rozpočtom a množstvom skúsených expertov. Preto si takáto spoločnosť môže dovoliť vykonávať náročné odhady pomocou metód, ako sú Delphi alebo metóda zdola nahor. Tieto sa jej oplatia hlavne pri veľkých dlhodobých projektoch, pretože presnejšími odhadmi si spoločnosť znižuje riziko neúspechu projektu. Takáto firma si tiež môže dovoliť prehodnocovať odhady počas vývoja, čo jej dáva výhodu.

Na druhej strane, malá začínajúca spoločnosť, ktorá nemá za sebou veľa projektov, musí zvoliť iný prístup. Takáto firma pravdepodobne nemá prostriedky ani skúsenosti na dlhý proces odhadovania. Preto je vhodné, aby sa použila metóda zhora nadol na začiatku

projektu a potom v priebehu projektu intuitívne alebo na základe predchádzajúcich skúseností odhadovať úsilie pre jednotlivé úlohy. Takýto prístup jej ale môže spôsobiť vážne problémy pri väčších projektoch, pretože pri nich sa nepresnosť odhadov prejaví výraznejšie.

Aj keď odhady nikdy nemôžu byť presné, stále sú dôležité pre proces plánovania projektu. Bez toho, aby sme niečo odhadli, nemôžeme naplánovať viac, ako jeden krok dopredu. Takže ak chceme vytvoriť plán, tak či chceme alebo nechceme, musíme vytvoriť aj nejaké odhady aj keby to bolo len intuitívne. Otázkou je, či je potrebné vynakladať špeciálne úsilie len na tvorbu odhadov. Myslím si, že sa to určite opláca, pretože presnejšie odhady pomáhajú k lepšiemu plánovaniu. Neznamená, že vždy treba robiť odhady tými najsofistikovanejšími metódami. Hlavne pri menších, menej zložitých projektoch, nie sú presné odhady také kľúčové ako pri veľkých, komplexných projektoch.

## Odhady a plánovanie v tímovom projekte

Tímový projekt, na ktorom v rámci štúdia pracujeme je dĺžkou trvania relatívne krátky, rovnako ako malý je aj tím, ktorý na ňom bude pracovať.

V našom tímovom projekte budeme na odhady používať úsudok experta, pričom ja budem prvý, kto zaujme túto úlohu. Na začiatku projektu som použil prístup zhora nadol, keď som odhadoval úsilie, ktoré bude na projekt potrebné. Nemeral som rozsah projektu, žiadnom parametrickou metódou. Využil som to, že skoro všetkých členov nášho tímu dobre poznám a aj som s nimi v minulosti spolupracoval. Neboli to vždy projekty takého rozsahu, ako je tento projekt, ale myslím si, že som sa naučil dosť o tom, ako členovia tímu pracujú a čo dokážu. Pri vytváraní plánu a určovaní úsilia pre jednotlivé úlohy som sa z časti riadil skúsenosťami z predchádzajúcich projektov. Konkrétne to je hlavne jeden vlastný projekt, na ktorom pracujem vo voľnom čase spolu ešte s ďalšími tromi kolegami. V tom projekte, zastávam asi všetky manažérske pozície a teda je mojou úlohou aj plánovanie a odhadovanie.

Dôležitá vlastnosť tímového projektu (možno len nášho zadania) je, že máme presne stanovený termín dokončenia, s ktorým sa nedá za žiadnych okolností hýbať, ale máme voľnosť pri tvorbe samotného produktu. Preto bude veľmi dôležité, aby sme počas špecifikácie a návrhu čo najlepšie odhadovali úsilie potrebné na jednotlivé vlastnosti a funkcie, ktoré by sme chceli do produktu zaradiť a navrhnuť ho tak, aby sme ho zvládli v zadanom čase dokončiť.

V momente, keď bude vypracovaný návrh, plánujem vytvoriť odhady metódou zdola nahor, pretože budem mať viac informácií o tom, čo treba v projekte vytvoriť. Budem môcť odhadnúť potrebné množstvo vynaloženého úsilia pre každý jednotlivý modul a tým pádom budem môcť lepšie prideliť úlohy.

Nebudeme používať metódu Delphi ani žiadnu inú zložitejšiu metódu, pretože pri projekte takéhoto rozsahu by to spotrebovalo príliš veľa úsilia a neposkytlo by nám to dostatočné veľkú výhodu, aby sa nám to oplátilo.

## Záver

Tvorba odhadov je dôležitou súčasťou plánovania softvérových projektov. Existuje viacero možností, ako ich vytvárať. Rôzne metódy majú rôznu presnosť výsledných odhadov a vyžadujú rôzne množstvo úsilia. Väčšinou platí, že čím presnejšia je metóda, tým je náročnejšia. Neexistuje žiadna metóda, ktorá je univerzálne vhodná pre všetky tímy a projekty. Napriek tomu je hociktorá metóda lepšia, ako obyčajné hádanie. Pri odhadovaní je vhodné použiť údaje z predchádzajúcich projektov, ale je potrebné zvážiť na koľko sú dané údaje relevantné pre aktuálny projekt. Pre veľké projekty je presnosť odhadov zvlášť dôležitá, preto je treba dať dôraz na proces odhadovania, inak sa vystavujeme vážnym rizikám.

Zaujímavou možnosťou pre ďalšiu prácu by bol prieskum možností lepšej reprezentácie znalostí potrebných na vytvorenie projektu a vyjadrenie rozsahu projektu pomocou tejto reprezentácie.

## Použitá literatúra

1. 2001. Estimating software projects. *SIGSOFT Softw. Eng. Notes* 26, 4 (Jul. 2001), 60-67.
2. Armour, P. 2002. Ten unmyths of project estimation. *Commun. ACM* 45, 11 (Nov. 2002), 15-18.
3. Peixoto, C. E., Audy, J. L., and Prikladnicki, R. 2010. The importance of the use of an estimation process. In *Proceedings of the 2010 ICSE Workshop on Software Development Governance* (Cape Town, South Africa, May 08 - 08, 2010). SDG '10. ACM, New York, NY, 13-17.

## Annotation

### *Accurate estimate? Impossible!*

*Estimation of software project can greatly influence its planning process, its course, and its outcome. How the estimate affects the project, depends on its accuracy. Generally, the more accurate the estimate, the better. But how do we obtain an accurate estimate? Is it even possible? Creating a good estimate of a software project is costly and difficult. Is it beneficial for all projects to create estimates? If yes, how should they be created? I look for answers for these and other questions in my essay. I address several methods for estimating software projects, their performance and accuracy. I address the need for estimating various types of software projects and how should these projects be estimated. Finally, I discuss the issue of estimating in the light on our team project.*