

# Sledovanie postupu softvérového projektu a manažment

PETER PRIKRYL

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
p.prikryl@gmail.com*

**Abstrakt.** Napriek najlepšiemu plánovaniu, účasti najlepšieho tímu a predvídaníu všetkých možných úskalí to vyzerá tak, že softvérové projekty majú dlhoročnú prax vo vyvolávaní nepredvídateľných problémov. Jedna z možností, ako sa môžeme vyhnúť neúspešnému ukončeniu projektu, je sledovanie postupu softvérového projektu. Aby sme vôbec mohli sledovať vývoj a postup v softvérovom projekte, je veľmi dôležité vytvoriť plán. Tiež je potrebné zaviesť vhodné metriky, ktorými budeme merať postup. Metriky by nemali nútiť ľudí dosahovať lepšie výsledky v meraní, ale motivovať ich efektívne dosiahnuť vytýčené ciele projektu. Keď máme plán projektu, je možné pýtať sa, akú časť projektového plánu sa podarilo splniť, odhadovať, koľko práce ešte treba spraviť a porovnávať naplánovaný stav so skutočným stavom. V prípade rozdielov je možné plán skorigovať a nasmerovať úsilie tímu tým správnym smerom.

## Úvod

Významným problémom v softvérovom projekte počas fázy vykonávania projektu je jeho kontrola a sledovanie. Napriek najlepšiemu plánovaniu, účasti najlepšieho tímu a predvídaníu všetkých možných úskalí to vyzerá tak, že softvérové projekty majú dlhoročnú prax vo vyvolávaní nepredvídateľných problémov.

Aby sa predišlo možnému neúspechu softvérového projektu, je dôležité sledovať charakteristiky projektu a vývoj týchto charakteristík v čase. Tým, že manažér vidí skutočný stav softvérového projektu, môže v prípade rozdielov medzi plánovaným a skutočným stavom projektu zasiahnuť a usmerniť vývoj ďalšieho postupu v projekte. Vyhodnocovanie vykonanej práce a analýza nákladov a úsilia vynaložených na vykonanú prácu napomáhajú efektívnejšiemu využitiu zdrojov v budúcnosti.

## Čo je to postup

Je jednoduché povedať, že niekto pracuje. Ale ako sa dá povedať, že niekto postupuje? Sledovať postup pri jednoduchej práci, ako napríklad maľovanie bytu, je jednoduché. Vždy po vymaľovaní jednej steny alebo miestnosti je postup viditeľný každému. Ale pri zložitejšej práci, akou je napríklad aj tvorba softvérového systému, je ťažšie identifikovať skutočný postup.

Niektoré činnosti vyžadujú premýšľanie a skúmanie, ktoré môže trvať hodiny alebo dni bez viditeľných výsledkov. Iná práca môže zahŕňať toľko rôznych podúloh alebo rozhovorov s ostatnými, že sa nedá žiadnym spôsobom zistiť, ako efektívne je táto práca vykonávaná, alebo či vôbec úsilie vynaložené na túto prácu povedie k postupu v softvérovom projekte. Pri zložitej práci alebo pri práci s veľkým množstvom ľudí je vždy ťažké rozoznať prácu v projekte od postupu v projekte.

Pre softvérový projekt je v prvom rade dôležité vytvoriť plán projektu. Plán nám umožňuje efektívne smerovať svoje úsilie na prácu, ktorú treba vykonať, aby sme splnili vytýčené ciele projektu. Keď máme plán projektu, je možné pýtať sa, akú časť projektového plánu sa podarilo splniť, odhadovať, koľko práce ešte treba spraviť a porovnávať naplánovaný stav so skutočným stavom. V prípade, že dosiahnuté výsledky nezodpovedajú naplánovaným, je však nutné zmeniť plán.

V tímoch, v ktorých je slabé riadenie, vzniká priestor pre to, aby bola vynaložená energia na zbytočnú alebo kontraproduktívnu prácu. Väčšinou je to spôsobené tým, že manažér si to nevšimne, nestará sa, alebo si nenájde čas, aby nasmeroval úsilie ľudí užitočnejším smerom. Keď sa niekto pustí do práce, nezáleží na tom, ako rýchlo v práci postupuje, ak postupuje nesprávnym smerom. Nasmerovanie úsilia môže byť oveľa dôležitejšie než veľkosť samotného vynaloženého úsilia.

## Ako určiť postup v softvérovom projekte

Nevyhnutnou súčasťou toho, aby sme mohli určiť, sledovať a kontrolovať postup softvérového projektu, je plánovanie a ohraničenie cieľov, ktoré sa majú pracou v softvérovom projekte dosiahnuť. Ak by som niekomu dal do ruky valček na maľovanie, ukázal mu tri kýble bielej farby a povedal mu, nech vymaľuje celý byt, bolo by jasné, ako má vyzeráť postup. Aj pri zložitejšej práci je úloha projektového vedúceho v tíme rovnaká – jeho úlohou je, aby každý vedel, ako má vyzeráť výsledok v čase, keď projekt skončí.

To, ako má na konci vyzeráť výsledok softvérového projektu, sa môže počas vykonávania projektu s časom meniť a je úlohou projektového vedúceho, aby sa táto predstava menila približne rovnako v celom tíme. Ak v tíme ešte nie je predstava o konečnom výsledku softvérového projektu, prvou úlohou, ktorú môže projektový vedúci v tíme zadať je, aby každý prispel názorom alebo nápadom, ako má konečný výsledok projektu vyzeráť.

Ďalšou súčasťou určovania postupu softvérového projektu je dennodenné riadenie tímu správnym smerom. Je úlohou projektového vedúceho tímu, aby neustále poukazoval na to, ako práca členov tímu prospieva splneniu vytýčených cieľov. Pri každom stretnutí musí projektový vedúci tímu motivovať ľudí tak, aby ich práca posúvala softvérový projekt bližšie k jeho vytýčeným cieľom, a teda aj bližšie k jeho úspešnému ukončeniu.

Projektový vedúci má najlepší prehľad o aktuálnom stave softvérového projektu. O svoj pohľad by sa mal čo najčastejšie deliť s ostatnými členmi tímu. Najlepšie vidí aj to, kedy ľudia svojou prácou postupujú nesprávnym smerom, alebo v horšom prípade znehodnocujú prácu iných členov tímu. Ak sa o svoj pohľad rozdelí s ostatnými členmi tímu, hoci nepozná odpovede na všetky otázky, ľudia jeho názor prijímú a upravia svoju prácu tak, aby maximalizovali postup v softvérovom projekte.

## Sledovanie práce v softvérovom projekte

Aby sme mohli sledovať prácu a postup v softvérovom projekte, potrebujeme ich vedieť najprv nejakým spôsobom zmerať. Predstava je taká, že zavedením metriky umožníme ľuďom sústrediť sa na dosiahnutie lepších výsledkov. Efekt takéhoto merania však často býva iný. Ľudia sa sústredia viac na dosiahnutie lepších výsledkov v meraní namiesto dosiahnutia vytýčených cieľov projektu.

V niektorých organizáciách merajú [3]:

- Koľko hodín strávil pracovník v kancelárii.
- Počet napísaných hlásení alebo špecifikácií.
- Počet prezentácií.
- Množstvo telefónnych hovorov.
- Počet napísaných riadkov kódu.
- Počet opravených chýb.
- Množstvo poslaných e-mailov.

Všetky merania využívajúce spomenuté metriky majú pomerne nízku výpovednú hodnotu, pretože zachytávajú iba vykonané aktivity. Napríklad, ak v organizácii, v ktorej pracujem, merajú počet opravených chýb, pre mňa to znamená, že budem opravovať hlavne tie jednoduchšie chyby. Stihnem ich opraviť viac a budem aj lepšie hodnotený. Chyby, ktorých oprava by trvala príliš dlho, v softvérovom projekte zostanú. Podobne aj meranie počtu riadkov v programe vypovedá iba o rozvlácnosti napísaného kódu, ale už nie o jeho kvalite. Každé z týchto meraní má svoje slabé miesta, ktoré sa dajú veľmi ľahko zneužiť. Ale keďže je také jednoduché ich zmerať, často sa v praxi používajú. Projektovému vedúcemu informácie z týchto meraní môžu pomôcť, ale len v prípade, ak sú interpretované s rozumom a k dispozícii sú aj ďalšie informácie.

## Sledovanie postupu v softvérovom projekte

Sledovanie postupu v softvérovom projekte už nie je také jednoduché ako sledovanie vykonávania práce. Spôsoby, akými sa meria postup v softvérovom projekte, sú viac subjektívne a vyžadujú viac rozmýšľania. Pre sledovanie postupu v softvérovom projekte je dobré klásť si nasledovné otázky o vykonávanej práci:

- Ku ktorému cieľu prispieva táto práca?
- Ako nám táto práca pomôže dostať sa bližšie k cieľu?
- Aký veľký je to príspevok k dosiahnutiu cieľa? (Dostane nás to bližšie k cieľu o 5 % alebo o 50 %?)
- Je kvalita vykonávanej práce dostatočná?
- Čo nám táto práca umožní ďalej vykonať? Ako nám pomôže v inej práci, ktorú treba tiež vykonať?

Zodpovedanie týchto otázok nám vždy pomôže utvoriť si obraz o postupe v projekte. Samozrejme záleží aj od toho, na akej úrovni podrobnosti sú vytýčené ciele definované. Ak sú ciele definované príliš nepresne, môže byť postupom aj práca, ktorá nemusí priniesť očakávané výsledky. Ak sú však ciele príliš konkrétne a sú definované príliš striktné, nezostáva priestor pre tvorivé riešenie problémov.

Určenie vhodnej úrovne podrobnosti pri definovaní cieľov je umenie a dá sa dobre zvládnuť iba na základe skúseností. Vo všeobecnosti je vhodné určiť ciele so vzrastajúcou úrovňou podrobnosti pre celý projekt, každý tím a ciele pre každého člena tímu.

## Pomer práce a postupu

Pri riešení nových problémov je často potrebné vyskúšať viacero návrhov a preskúmať viacero alternatív. Treba nájsť nové spôsoby riešenia alebo skombinovať existujúce myšlienky. Je to proces skúmania a experimentovania. Tvorivá práca je neefektívna. Treba pri nej dlho skúmať a experimentovať, aby sme nakoniec dosiahli nejaký postup.

Projektový vedúci potrebuje určiť, aký pomer práce k postupu pre splnenie daného cieľa sa očakáva. Môže tak pomôcť ľuďom oddeliť úlohy, ktoré budú vyžadovať veľa pokusov, aby ich bolo možné splniť (napríklad návrh novej architektúry softvérového systému) od tých, ktoré sú priamočiare a nevyžadujú veľa rozmýšľania (napríklad naplnenie databázy).

Pomer práce k výkonu vyjadruje, koľko práce (alebo času) je potrebnej pre dosiahnutie jednotky postupu. Napríklad [3]:

- Naplnenie databázy údajmi: 1 k 1.
- Návrh nového vyhľadávacieho algoritmu: 5 k 1.

– Pretvorenie celého používateľského rozhrania: 5 k 1.

– Vytvoriť perpetuum mobile:  $\infty$  k 1.

Manažér môže ľuďom pomôcť, aby ich pomer práce k postupu bol čo najnižší tým, že sa bude po každej úlohe pýtať tieto otázky:

– Čo sme sa naučili z tejto úlohy, čo by nám mohlo pomôcť s ďalšou úlohou?

– Sú nejaké nové otázky k ďalšej úlohe, ktoré by zvýšili veľkosť postupu?

– Potrebujeme ďalšie informácie, ktoré zlepšia ďalší pokus?

– Dajú sa ciele definovať lepšie?

Pomer práce k postupu naznačuje ešte ďalšiu vec: Pracovník, ktorý je skúsenejší alebo šikovnejší, dosiahne lepší pomer práce k postupu, než dosiahne menej skúsený pracovník.

Z pomeru práce k postupu je zrejmé, že rôzna úroveň kvality bude mať rôzny pomer práce k postupu. Najst' dostatočné riešenie môžeme už po piatich pokusoch, ale až po pätnástich pokusoch môžeme najst' výborné riešenie. Úlohou projektového vedúceho je teda pomôcť tímu rozhodnúť, aké veľké investície sú vhodné pre dosiahnutie daných cieľov.

## Úskalia sledovania postupu v softvérovom projekte

V nástrojoch na riadenie projektu, akým je napríklad aj Microsoft Project, existuje v niektorých prípadoch klam, ktorý hovorí, že zadaná úloha môže byť čiastočne splnená (napr. že úloha môže byť splnená na 10 % alebo 20 %). Takýto stav vzniká pri automatickom dopočítaní stavu úlohy na základe aktuálneho času a času vyhradeného na úlohu.

Ak pod úlohou myslíme splnenie nejakého cieľa, tento klam je očividný – buď cieľ splnený je, alebo cieľ splnený nie je. Je to čierno-biela, binárna logika. Ak je cieľ definovaný neurčito a formulácia zámeru je nepresná, ako napríklad „napíš nejaké vyjadrenie k posudku“, tak je táto úloha splnená v momente, keď ju začneme vykonávať. Akonáhle napíšeme jednu vetu, napísali sme „nejaké vyjadrenie k posudku“.

Na druhú stranu, ak je úloha dobre definovaná a jej výsledky sa dajú odovzdať, potom dosiahneme splnenie úlohy práve vtedy, keď odovzdáme jej výsledok. Napríklad „vytvor kompletnú používateľskú príručku a technickú príručku“. Táto definícia úlohy je oveľa užitočnejšia, pretože jej splnenie sa dá jasne zmerať. Úloha je splnená, iba ak sú dokumenty napísané, boli skontrolované a opravené a sú pripravené pre zverejnenie. Skončili sme, ak sa už nemusia robiť ďalšie zmeny a môžeme sa presunúť na ďalšiu úlohu.

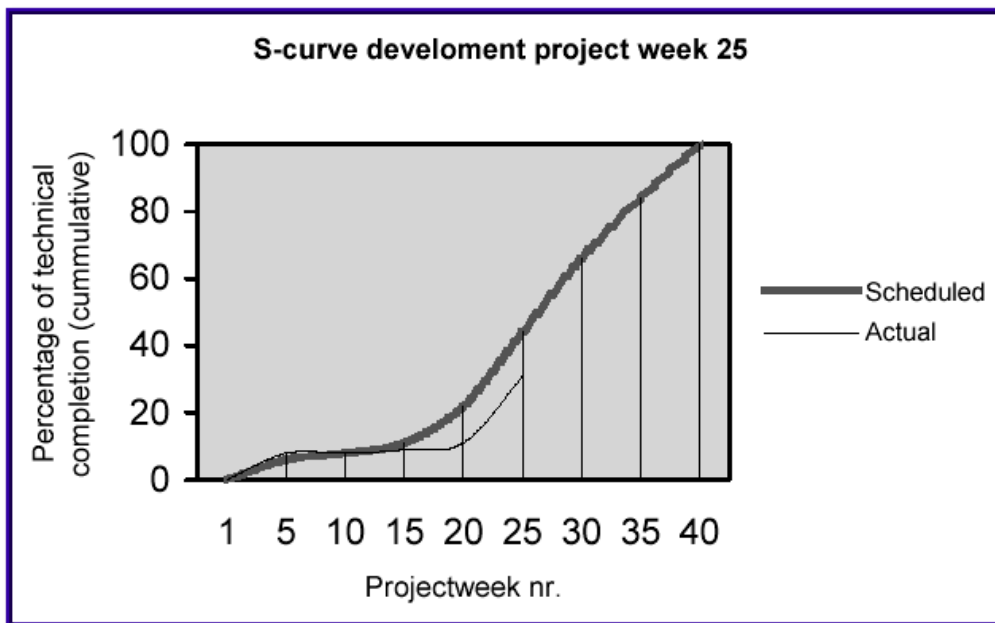
Nebezpečenstvo toho, že uveríme, že úloha môže byť čiastočne splnená je, že dáva falošný pocit bezpečia. Ľudia budú tvrdiť, že splnili úlohu na 50 % v polovici jej vyhradeného času, pretože definovať, ako vyzerá úloha splnená na 50 % môže byť náročné. Aj v prípade, že ešte treba dokončiť 90 % úlohy, budú stále trvať na tom, že

keďže prešla polovica času vyhradeného na danú úlohu, musela zostať tiež už iba polovica práce, ktorú treba vykonať. Túto mylnú predstavu majú obzvlášť ľudia, ktorí veria, že čas je elastický. To znamená, že sa dá nahustiť ľubovoľné množstvo práce na danú dĺžku času, záleží iba na tom, ako tvrdo sa pracuje [1]. Ale v softvérových projektoch často platí, že keď sa zdá, že je 90 % úlohy hotovej, zostáva ešte dokončiť zvyšných 90 %.

## Zobrazenie postupu v softvérovom projekte

Je dôležité, aby tím pracujúci na projekte, ale aj samotný zákazník videli postup v softvérovom projekte. Výsledky v softvérovom projekte majú slabšiu viditeľnosť a väčšinou je v projekte zapojených veľa ľudí z rôznych oblastí. Preto treba pri zobrazovaní postupu zvoliť jednoduchú metriku. Tá sa dá zvoliť analýzou práce, ktorú bude treba vykonať. Práca sa potom rozdelí na menšie úlohy a ohodnotí sa ich príspevok k celkovému postupu. Po rozdelení práce a jej naplánovaní máme predstavu o tom, ako bude vyzeráť vývoj projektu.

Naplánovaný stav projektu môžeme zakresliť do grafu, kde na x-ovej osi bude čas a na y-ovej osi bude vykonaná práca. Pod vykonanou prácou môžeme rozumieť napríklad počet hotových tabuliek v databáze alebo počet naprogramovaných a otestovaných funkcií v projekte. Do tohto grafu sa bude časom zakresľovať aj skutočný stav projektu. Postup sa nesleduje neustále, ale v pravidelných intervaloch.



**Obr. 1.** Graficky znázornený naplánovaný a skutočný vývoj projektu [2].

Z obrázku **Obr. 1**, ktorý zachytáva vývoj stavu (teda postup) v softvérovom projekte, sa dajú prečítať nasledovné skutočnosti:

- Z horizontálneho posunu naplánovaného a skutočného stavu projektu sa dá usúdiť, že projekt má dvojtýždňové meškanie.
- V projekte bolo desať týždňov, počas ktorých v projekte nenastal žiadny merateľný postup.
- Koniec naplánovanej krivky v posledných týždňoch je príliš strmý. To naznačuje, že dĺžka trvania projektu sa výrazne predĺži. Pretože v súčasnom týždni (25) je krivka so skutočným stavom paralelná s naplánovanou krivkou, je pravdepodobné, že sa bez zásahu do plánu nestihne dohnať dvojtýždňové meškanie.

Toto nie je jediný spôsob zobrazenia postupu v softvérovom projekte. Často sa používa aj graf, ktorý je opačne orientovaný – zachytáva, koľko toho ešte treba urobiť, namiesto toho, koľko sa už urobilo. To, ktorý z nich sa zvolí, závisí iba od psychológie. Nieкого viac motivuje, keď sa hýbe po krivke smerom nahor, ďalší chce dosiahnuť nulu.

## Záver

Ak chceme vidieť v softvérovom projekte nejaký postup, nezaobídeme sa bez plánu. Plán nám umožní sledovať postup v softvérovom projekte. Pomocou plánu vieme zhodnotiť, aký mal byť skutočný stav projektu a ako veľmi sa líši od naplánovaného stavu projektu. Sledovanie vývoja charakteristík softvérového projektu v čase vykonávania projektu umožňuje projektovému vedúcemu riadiť projekt a v prípade rozdielov od naplánovaného stavu včas zasiahnuť a nasmerovať ľudí správnym smerom. Vyhodnocovanie vykonanej práce a analýza nákladov a úsilia vynaložených na vykonanú prácu napomáhajú aj efektívnejšiemu využitiu zdrojov v budúcnosti.

Pri sledovaní softvérového projektu je pritom dôležité rozoznať prácu od skutočného postupu. Zákazníka väčšinou nezaujíma, že na projekte sa pracuje, skôr ho zaujíma, aký bol postup: Čo sa od posledného stretnutia zmenilo, ako sa tím priblížil k dosiahnutiu dohodnutých cieľov. Aj preto je dôležité, aby sme vedeli postup nejakým spôsobom zmerať, prípadne zakresliť do grafu. Zvolený graf má aj psychologický efekt, môže motivovať k ďalšej práci a ďalšiemu postupu. Okrem aktuálneho stavu sa z neho dajú rýchlo vyčítať napríklad aj možné problémy a môže byť aj podnetom pre zmenu plánu.

## Použitá literatúra

1. Nick Jenkins: *A Project Management Primer or "a guide on how to make projects work"*, 2005.  
<http://www.nickjenkins.net/prose/projectPrimer.pdf>

2. Joop van der Linden: *Monitoring Progress in Software Development*, júl 2003.  
<http://www.stsc.hill.af.mil/crosstalk/2003/07/vanderLinden.html>
3. Scott Berkun: *Work vs. Progress*, august 2005.  
<http://www.scottberkun.com/essays/essay45.htm>

## **Annotation**

### *Monitoring progress in software project and management*

Despite the best planning, the best team participation and expecting all possible pitfalls, it seems that software projects are experienced to cause unexpectable problems. One possible way how to avoid unsuccessful project termination is to monitor progress of the software project. To be able to monitor development and progress of the software project it is important to create a plan. It is also needed to choose feasible metrics to measure the progress. Choosed metrics should not drive people to achieve better results in a measurement, but it should motivate them to effectively attain identified goals of the project. When the plan is ready, it is possible to ask, which planned tasks are already accomplished, estimate amount of tasks remaining and compare scheduled project status to the real project status. In case of difference, it is possible to correct the plan and aim the effort to the right direction.