MONITOROVANIE, CESTA K ÚSPECHU?

Časový termín niekedy alebo vždy nestihne podľa prieskumu až vyše 80% opýtaných.

Miroslav Kacera

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
kacera.m[zavináč]zoznam[.]sk

Abstrakt. *Pred štartom akéhokoľvek projektu sa vypracuje plán, ktorý má viesť k jeho úspešnému ukončeniu. Tá ťažšia časť je tento plán následne dodržať. Pri tvorbe softvérových produktov sa manažéri stretávajú s mnohými problémami, ktoré ohrozujú úspešné dokončenie produktu. Ak sa produkt aj podarí dokončiť, často je to za cenu prekročenia rozpočtu, alebo nedodržania časového harmonogramu. Väčšine z týchto problémov by sa však dalo predísť, keby manažér dokázal dostatočne skoro odhaliť, že vývoj sa uberá nesprávnou cestou. Ako to ale včas zistiť? Odpoveďou je monitorovanie stavu projektu. Ide o mechanizmus, pomocou ktorého sa na základe analýzy zozbieraných údajov vyhodnotí, v akom stave sa projekt nachádza. Následne, ak je to potrebné, sa aplikujú zmeny, ktoré by mali viesť k ozdraveniu projektu. Je však samotné monitorovanie naozaj zárukou úspechu?*

Kľúčové slová: *monitorovanie, manažment, softvér, projekt, pokrok, sledovanie*

Na ceste k monitorovaniu

Vývoj softvérových produktov je pomerne mladá disciplína, avšak od jej vzniku prešla veľkými zmenami. Spočiatku boli programy vyvíjané len jednotlivcami, respektíve malou skupinou ľudí. Avšak s rozvojom osobných počítačov, sietí a internetu, sa menili aj požiadavky na vyvíjaný softvér. Požiadavky sú dnes komplexnejšie a rozsiahlejšie a nie je možné, aby jeden človek dokázal takýto softvér vytvoriť v požadovanom rozsahu, kvalite a v relatívne krátkom čase.

Pre potreby vývoja takýchto softvérových produktov začali vznikať veľké tímy programátorov, analytikov, dizajnérov a iných odborníkov, ktorí sa zúčastňujú na jeho tvorbe. Je samozrejmé, že takýto tím ľudí by nemohol byť úspešný bez potrebného vedenia. Vedenie tímu majú na starosti manažéri, ktorých úlohou je, aby bol produkt vytvorený v požadovanej kvalite a načas.

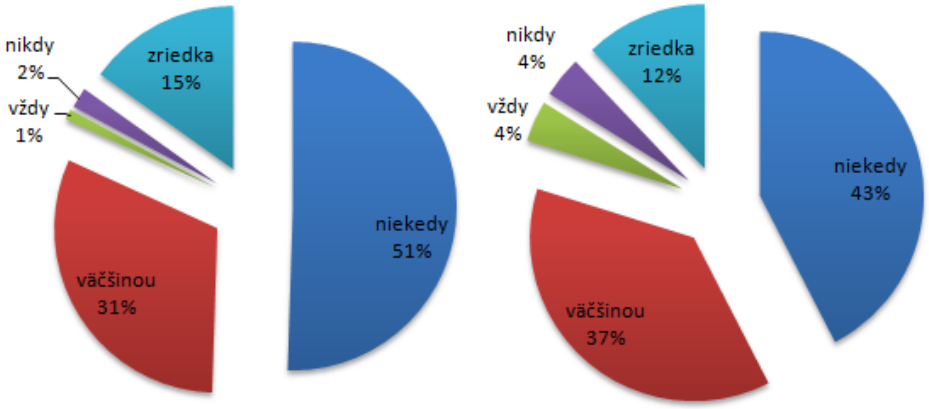
Potrebu vedenia a manažovania nie je vidieť len pri vývoji softvérových produktov, ale aj v iných oblastiach. Už oddávna každá väčšia skupina ľudí snažiaca sa dosiahnuť nejaký cieľ (tím) mala svojho lídra (manažéra). Vhodným príkladom je šport. Každý športový tím má svojho manažéra - trénera, ktorý na základe *monitorovania* robí zmeny, ktoré vedú k lepším výsledkom. Z histórie sú to napríklad vojská a ich velitelia, atď.

Je potrebné monitorovať vývoj softvéru?

Úlohou tímu je vytvoriť *v stanovenom čase* čo najkvalitnejší produkt a neprekročiť *stanovený rozpočet*. Tím pozostáva z rôznych osobností, ktorých skúsenosti a znalosti sa môžu líšiť. Takisto aj ich motivácia je rôzna.

Keby vývoj produktu nikto neviedol, tak je zrejmé, že drvivá väčšina projektov by nebola úspešná. Ak by aj produkt vznikol, zrejme by nebol dostatočne kvalitný, alebo by bol dokončený až po stanovenom termíne. Je však samotné vedenie zárukou úspechu?

Podľa prieskumu Phana, Vogela a Nunamakera [4], len 16% respondentov uviedlo, že nikdy alebo len málokedy mali problémy s prekročeným rozpočtom. Na opačnej strane, až 37% respondentov uviedlo, že takéto problémy majú skoro vždy. Časový termín niekedy alebo vždy nestihne podľa prieskumu až vyše 80% opýtaných. Podrobnejší prehľad výsledkov prieskumu poskytujú grafy na obrázku Obr. **1**.



Obr. 1. Graf rozloženia neskorého doručenia (vľavo) a prekročenia rozpočtu (vpravo) [4].

Aj z vlastnej skúsenosti viem, že úsilie stúpa väčšinou až s blížiacim sa termínom dokončenia. To často platí aj pri projektoch, keď sa zdržania vo vývoji neriešia priebežne, ale až v neskorých fázach projektu. Vtedy sú korekcie oveľa ťažšie a majú väčšie následky [5]. V následnej snahe zvládnuť toho čo najviac v čo najkratšom čase, sa znižuje kvalita riešenia a často sa tak viac času stratí, ako sa získa. Je preto dôležité, aby sa *priebežne monitoroval* stav projektu a pokiaľ má projekt meškanie, je treba prehodnotiť plán

a s chladnou hlavou nájsť riešenie, ktoré bude reálne uskutočniteľné. Dôsledky neodhalenia chýb v jednotlivých fázach projektu veľmi prudko rastú, čo potvrdzuje aj tabuľka nižšie. Pritom pri veľkých projektoch, ktoré pozostávajú zo stoviek úloh a pracujú na nich stovky ľudí, je šanca na bezproblémový priebeh projektu menšia ako na výhru v lotérii. Z tohto vyplýva, že monitorovanie projektu je nielenže potrebné, ale pre úspech veľkých projektov, podľa môjho názoru, *nevyhnutné*!

Tab. 1. Náklady na opravu chyby v priebehu projektu [2].

ETAPA	NÁKLADY (človeko-hodiny)
Špecifikácia	2
Návrh	5
Implementácia	15
Akceptačné testovanie	50
Údržba	150

Čo sa môže pokaziť

Monitorovanie projektu vyzerá na prvý pohľad jednoducho. Človek si povie, že na tom predsa nemôže byť nič ťažké. Ako sa hovorí: „Pozriem a vidím“. Avšak, aby monitorovanie prinieslo aj nejaký skutočný úžitok, musí sa k nemu pristupovať zodpovedne. Nesprávne monitorovanie môže mať horšie následky ako žiadne.

Podľa môjho názoru, azda najdôležitejšie je, aby sa *stav projektu hodnotil čo najčastejšie*. Ak sa napríklad urobí nejaká chyba na začiatku mesiaca a odhalí sa až pri hodnotení na jeho konci, je potrebné vynaložiť väčšie úsilie na jej odstránenie, ako keby bola odhalená po týždni. Na druhej strane na monitorovaní sa podieľa veľký počet ľudí a generuje sa veľký počet hlásení, a preto by každodenná kontrola stála veľa času, čo by v konečnom dôsledku projekt spomaľovalo. Existujú síce aj také prístupy, pri ktorých sa organizujú každodenné stretnutia tímu (agilné programovanie), ale podľa môjho názoru tento prístup nie je vhodný pre veľké tímy a ide skôr o stretnutia v rámci časti tímu. Tiež si myslím, že tento prístup nemôže zastúpiť monitorovanie v plnom rozsahu. Ako teda nastaviť frekvenciu monitorovania tak, aby sme mali projekt pod kontrolou a zároveň nespomaľovali zbytočne jeho vývoj? Podľa Elizabeth Clark sú mesačné správy príliš zriedkavé na to, aby monitorovanie plnilo svoju úlohu. Za najvhodnejšie riešenie považuje týždenné správy [5].

Samotný proces monitorovania pozostáva z troch hlavných častí.

- získanie informácií o stave projektu a ich vyhodnotenie,
- porovnanie stavu s plánom,
- aktualizácia plánu, ak je to nutné.

Z môjho pohľadu je najnáročnejšou úlohou získavanie potrebných informácií. Tieto informácie môžu byť ľahko *chybné a skreslené*.

Aby bola dosiahnutá čo najvyššia efektivita monitorovania, je dôležité, aby manažéri zbierali len také dáta, ktoré sú použiteľné. Dáta netreba zbierať len preto, lebo sú k dispozícii, ale preto, že sú *užitočné*. Nepodstatné dáta zbytočne spomalia proces

monitorovania a neprinesú žiadne výsledky [3]. Čo však považujem za väčší problém, je skreslenosť zozbieraných dát. Ide o to, že žiadny riaditeľ alebo manažér nerád počúva zlé správy, čím vytvára miesto pre „prikrášľovanie" stavu projektu. Keďže väčšina ľudí pracujúcich na projekte hodnotí svoju vlastnú prácu (alebo prácu, za ktorú sú zodpovední), je prirodzené, že ju ohodnotia o niečo *lepšie, ako to v skutočnosti je*. To navodzuje dojem, že projekt ide podľa plánu, čo však v skutočnosti nemusí byť pravda. A o tom, ako sa takéto skreslenie prejaví v neskorších fázach projektu, som písal v predchádzajúcej časti.

Tento problém, ako je vidieť, je skôr psychický ako technický a je hlboko zakorenený vo väčšine ľudí. Preto ho nie je jednoduché odstrániť. Ako riešenie vidím školenie vedúceho personálu, aby dokázali správne motivovať ľudí, hodnotiť svoju prácu reálne. Ak budú manažéri trestať každého, kto podá čestné, aj keď neradostné dáta, s veľkou pravdepodobnosťou bude ich projekt trpieť práve týmto problémom, čo v konečnom dôsledku spôsobí, že meškanie sa odhalí príliš neskoro a projekt skončí neúspechom. Na druhej strane je ale samozrejmé, že ak si daný človek riadne neodvedie svoju prácu, je nutné vyvodiť zodpovedajúce dôsledky.

Keď už sú všetky informácie úspešne zozbierané, treba ich vyhodnotiť. To takisto nie je taká ľahká úloha, ako by sa mohlo zdať. Existuje viac druhov prístupov, ale ani o jednom sa nedá povedať, že je najlepší. Najčastejšie dva prístupy, ktoré sa používajú na sledovanie pokroku (z ang. slova progress) projektu, sú založené na aktivite a na splnených jednotkách. Na prvý pohľad sa zdá, že obidva prístupy sú postavené na veľmi podobnom princípe a aj vzťah medzi aktivitou a splnenými jednotkami je pomerne viditeľný a ľahko definovateľný. Veď predsa čím viac jednotiek je hotových, tým väčšiu aktivitu sme museli vyvinúť a naopak. V čom je teda rozdiel? Môže sa stať, že máme projekt v určitej fáze vývoja a aplikujeme spomínané dva prístupy, no dostaneme úplne odlišné výsledky. Jeden hovorí, že projekt má výrazné meškanie a podľa druhého sme takmer pred dokončením. Ako je to možné? Pri hodnotení aktivity je výhoda, že každá odvedená práca sa premietne do stavu projektu. Avšak hrozí, že všetky časti projektu sú *takmer hotové* a pritom ešte nie je ako celok *nič funkčné*. Naopak, jednotky presne určujú, koľko sa z projektu urobilo a koľko je ešte nutné urobiť. Tu ale nie je započítaný žiadny kredit za prácu na ešte nedokončených jednotkách, čo sa na papieri prejavuje ako meškanie projektu. Zložitosť jednotiek sa tiež môže veľmi líšiť a ak sa na začiatku spravia najskôr tie ľahšie, vyzerá to, že projekt napreduje rýchlejšie ako sa plánovalo [5]. Napriek využívaniu spomínaných prístupov, hodnotenie pokroku je stále do veľkej miery ovplyvnené subjektívnym odhadom pozorovateľa.

Prečo je pokrok subjektívny?

Ohodnotiť, či niekto pracuje, je pomerne jednoduché, ale je možné automaticky prehlásiť, že zároveň robí pokrok? Ak kosíme trávnik alebo natierame steny, je to vcelku jednoznačné. Na prvý pohľad môžeme povedať, že každým prejdením štetca po nenatretej stene sme o niečo bližšie k dokončeniu. Ale pri komplexnej činnosti, v rámci veľkého počtu ľudí podieľajúcich sa na výsledku, to nie je také jednoznačné. *Práca nemusí automaticky znamenať pokrok*. Medzi takéto činnosti patrí aj tvorba softvérových produktov [1].

Práve preto neexistuje univerzálne meradlo pokroku. Všetko záleží od konkrétnej situácie. Veľmi jednoduchým príkladom toho, ako môže byť jedna a tá istá vec chápaná rôznym spôsobom, je vývoj funkčného a nefunkčného softvéru. Ak je požiadavka na nefunkčný program, tak každý zlý riadok kódu môže byť chápaný ako pokrok. To sa samozrejme nedá povedať, keď máme vytvoriť funkčný program. V praxi je tento príklad skôr úsmevný, ale existujú aj reálnejšie príklady. Keď máme vykopať jamu alebo ju naopak zasypať. Vykopanie jednej lopaty zeminy je tu chápané na jednej strane ako pokrok, na druhej ako kontraproduktívna práca.

Z toho vyplýva, že na to, aby sa minimalizoval dopad nesprávnych odhadov na projekt, je potrebné ohodnotiť stav projektu čo najpresnejšie. Ako ale môže manažér vedieť, či je jeho odhad kvalitný alebo nie? Predsa každý jeden manažér má záujem na tom, aby projekt skončil úspechom. Ide aj o jeho osobné záujmy a o každom odhade, ktorý urobí, je pravdepodobne presvedčený, že je správny. To však vždy ukáže až čas. A keďže každý odhad je svojím spôsobom špecifický, nie je možné takúto „zručnosť" získať teoretickým učením. Odhad sa väčšinou spresňuje *s rastúcimi skúsenosťami* manažéra. To znamená, že jediná cesta vedie cez učenie sa na vlastných chybách. Chyby ale majú v takýchto prípadoch často ďalekosiahle následky. Preto považujem takýto prístup za nevhodný. Čo vidím ako možné riešenie, je učiť sa z chýb iných. Myslím si, že pre firmy by bolo výhodnejšie zamestnávať v určitom období dvoch manažérov na tú istú prácu, ako sa spoliehať na odhady neskúseného manažéra. Menej skúsený alebo začínajúci manažér by sa mohol popri tom skúsenejšom natrénovať, aby neskôr dokázal lepšie odhadovať vzniknuté situácie.

Úloha manažéra je v konečnom dôsledku pomerne jasná. Viesť tím ľudí a na základe monitorovania organizovať podriadených tak, aby bol projekt úspešný. Práca manažéra môže výraznou mierou prispieť k tomuto úspechu, ale tak isto môže celý projekt potopiť. Ak manažér nevenuje vedeniu ľudí dostatočnú pozornosť a priebežne nesleduje prácu na projekte, môže sa stať, že aj skúsený tím zlyhá. Vývojári pracujúci na svojich úlohách si môžu myslieť, že robia pokrok, ale v skutočnosti ich práca môže byť neužitočná (napríklad dvaja vývojári robia to isté, alebo robia niečo, čo nie je potrebné). Manažér tu je na to, aby ľudí nasmeroval tým správnym smerom [1].

Neprehľadnosť monitorovania

Údaje z monitorovania pri veľkých projektoch sú často veľmi rozsiahle a neprehľadné. Manažéri musia bojovať s veľkým množstvom hlásení, tabuliek a ďalších údajov, ktoré dostanú od lídrov jednotlivých tímov. Pri takom množstve údajov je *ľahké prehliadnuť nejaké problémy* v projekte. Preto je nutné pri zavádzaní monitorovania myslieť aj na to, aby sa údaje dokázali relatívne jednoducho a rýchlo spracovávať. Najčastejšie sa používajú jednotné formuláre, tabuľky, alebo iné, pre firmu štandardizované prostriedky.

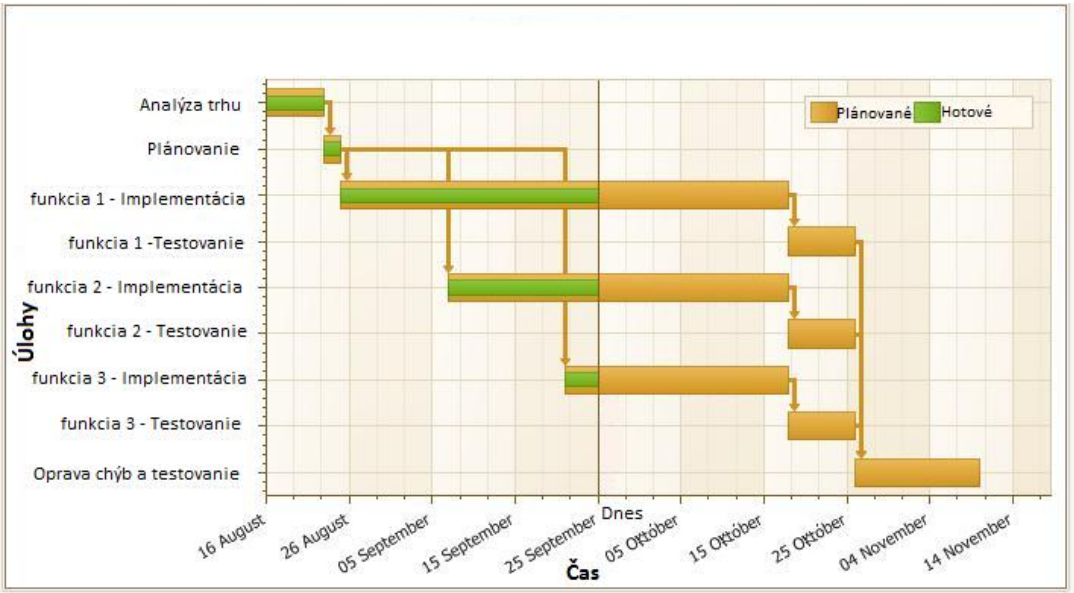
Ďalšia vec je reprezentácia získaných údajov. Zozbierané údaje samy o sebe nehovoria takmer nič o reálnom stave celého projektu. Ide len o akési fragmenty, ktoré je nutné zhromaždiť a vytvoriť z nich ucelený obraz.

Celý projekt pozostáva z množstva úloh, ktoré môžu bežať simultánne, alebo sú na sebe závislé a začínajú jedna po druhej. Pri monitorovaní chceme mať samozrejme prehľad

o všetkých úlohách, ktoré sa vykonávajú, boli vykonané, alebo ešte len čakajú na svoje začatie.

Je dôležité, aby sa definície jednotlivých úloh neprekrývali. To by potom spôsobovalo duplicitu pri vývoji a zbytočne by sa vyhodili prostriedky aj čas. Správne identifikovať jednotlivé úlohy teda nie je vždy úplne triviálnou úlohou. Preto existujú prístupy a nástroje, ktoré pomáhajú riešiť tieto problémy. Jedným z nich je Ganttov diagram. Pomocou tohto nástroja je možné vytvoriť ľahko zrozumiteľné vizuálne reprezentácie jednotlivých úloh v rámci celého projektu. Prehľadne zobrazuje, kedy úlohy začali, kedy majú skončiť, na koľko percent sú dokončené a aké zdroje sa na danú úlohu používajú. Manažér tak namiesto veľkého množstva abstraktných údajov má možnosť vidieť stav projektu zrozumiteľne a v celku. Ak manažér vidí, že niektorá úloha je oproti plánu v časovom sklze, môže vyčleniť na túto úlohu ďalšie zdroje.

Na obrázku nižšie je znázornený príklad Ganttovho diagramu pre nejaký hypotetický projekt. Je na ňom znázornený plán projektu a jeho aktuálny priebeh. Zelenou farbou je zobrazená tá časť úloh, ktorá už prebehla do dnešného dňa. Z diagramu sa dá vyčítať, na koľko percent je daná úloha splnená a koľko z nej ešte treba urobiť.



Obr. 2. Príklad Ganttovho diagramu [5].

Ako vidieť aj z obrázka, v prípade Ganttovho diagramu ide skôr iba o vizuálnu pomôcku, ktorá poskytuje náhľad nad projektom, a z ktorej sa dá ľahko identifikovať, na koľko percent sú jednotlivé úlohy rozpracované. Avšak to sú všetky informácie, ktoré tento diagram poskytuje. Nedozvieme sa z neho, koľko riadkov kódu je napísaných alebo koľko je urobených testov [5]. Tieto informácie musia byť na základe monitorovania do diagramu doplnené.

Záver

Vyvíjať softvérový produkt je náročná úloha, ktorá ak má byť úspešne zvládnutá, vyžaduje kvalitné plánovanie. Prax však ukazuje, že nie je jednoduché vytvoriť časový a rozpočtový plán pre veľké a zložité projekty. Nehovoriac o tom, že následne splniť takýto plán, navrhnutý pred začatím vývoja, je takmer nemožné.

Sledovať dodržiavanie plánu a pokrok projektu je úlohou monitorovania. V tejto eseji som poukázal na niektoré problémy, ktoré sa pri monitorovaní môžu vyskytnúť a tiež na ich možné riešenie.

Napriek existencii rôznych prístupov a postupov, ktoré pomáhajú monitorovať softvérový projekt, nie je jeho úspech nikdy na sto percent zaručený. Ako poukazuje táto esej, je to do veľkej miery kvôli tomu, že pokrok veľkých a komplexných projektov nie je jednoduché merať a odhad ich stavu je do veľkej miery založený na subjektívnych odhadoch manažérov a jednotlivých členov tímu.

Monitorovanie môže niekedy priniesť rozporuplné výsledky, ale v kombinácii so skúseným tímom, dobrým vedením a fungujúcou komunikáciou medzi členmi tímu, ide o nástroj, ktorý dosahuje reálne výsledky a veľkou mierou môže prispieť k zlepšeniu procesu vývoja softvérových produktov.

Použitá literatúra

1. Berkun, S.: *Work vs. Progress* [online]. 15. augusta 2005 [cit. 6. 9. 2009]. Dostupné na: <http://www.scottberkun.com/essays/45-work-vs-progress/>
2. Bielikova, M.: *Softvérové inžinierstvo*. Vydavateľstvo STU, Vazovova 5, Bratislava, 2000.
3. Buntrock, A.: *Progress & Performance Measurement* [online]. CMS Inc., 2002 [cit. 6. 9. 2009]. Dostupné na: <http://www.cms-inc.ca/learning/papers/prog_perform.html>
4. Genuchten, M.: Why is Software Late? An Empirical Study of Reasons For Delay in Software Development. *IEEE Transactions on Software Engineering*, Vol. 17., No. 6 (1991), 582 - 590.
5. International Function Point Users Group: *IT Measurement: Practical Advice from the Experts*. Addison-Wesley Professional, 2002, 223-236.

Annotation

Monitoring, the path to success?

Before the start of any project, there is a plan to develop. The plan which should led to project's successful completion. The difficult part is to follow such plan. When creating a software product, managers are facing many problems that threaten the successful completion of the product. Even if the product is completed, it often overruns budget, or it is developed after the deadline. Most of these problems could have been avoided if the manager had been able to detect that the development is heading the wrong way in time. But how to find it out? The answer is to monitor the project's status. Monitoring is a mechanism which evaluates the status of the project, based on the data

collected. Then, if necessary, the changes are applied, which should lead to the recovery of the project. However, does the monitoring itself really guaranteeing success?