

Sledovanie softvérového projektu a jeho vplyv na plánovanie a riadenie

PAVOL VRBA

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
vrba02@student.fiit.stuba.sk*

Abstrakt. Počas tvorby softvérového projektu musí manažment určitým spôsobom sledovať jeho vývoj. Riešiť problémy rýchlosti vývoja, plánovania, nákladov a riadenia pri postupovaní k úspešnému zakončeniu projektu. Nie je ničím výnimočným, keď sa projekt oneskorí alebo predraží. Úlohou dobrého manažmentu je zabezpečiť priebežné sledovanie projektu. Manažment by mal zhotoviť a zaviesť systém na sledovanie stavu projektu a kontrolný plán. Zo získaných hodnôt určiť stav a prispôsobiť proces vývoja, čo sa týka plánovania a riadenia. Takéto automatické sledovanie môže manažérov včas varovať na prichádzajúce problémy a umožňuje im tak vykonať náležité opatrenia. Táto práca opisuje niektoré z techník používaných pri sledovaní softvérového projektu.

Úvod

Sledovanie softvérového projektu je dôležitá činnosť počas vývoja softvéru. Jeho účelom je zaistiť viditeľnosť v postupe tak, aby manažment mohol včas podniknúť opravné opatrenia, keď sa rýchlosť vývoja viditeľne odchýli od plánu. Medzi takéto opatrenia patrí napríklad časté preplánovanie a prerozdelenie úloh. Preplánovanie znamená zmeny v projektovom plánovaní a vývoji, ktoré by mali byť vykonané a riadené.

Je teda zrejmé pre každého projektového manažéra, že sledovanie postupu je dôležité. Ťažšie je určiť spôsob sledovania, ktoré by mu pomohlo robiť adekvátne rozhodnutia, na základe ktorých by strávil čas riešením dôležitejších problémov, ktoré by mohli urýchliť vývoj.

Postup

Je vcelku ľahké povedať, že niekto pracuje. Už ale nie je také jednoduché povedať, že niekto postupuje. Jednoduché činnosti ako napríklad kosenie trávnikov alebo umývanie

auta majú celkom viditeľný stav postupu pre každého. Aj nezainteresovaný vidí, koľko sa stihlo spraviť a koľko práce ešte zostáva. Pri komplexnejších činnostiach ako napríklad tvorba softvéru alebo riadenie väčšieho podniku, už postup nie je taký viditeľný a je celkom zložitý ho naozaj správne určiť. Zložitá práca alebo práca väčšieho počtu ľudí na jednom projekte znamená ťažšie oddeliť prácu od postupu.

Jedným z problémov postupu je, že je subjektívny. Žiaľbohu neexistuje univerzálny ideálny spôsob na posúdenie, či sa veci hýbu alebo nie. V zle riadených projektoch sa môže stať, že niekto pracuje na nepodstatnej veci a projektový manažér si toho nemusí všimnúť. Schopní ľudia tak môžu pracovať na veciach, o ktorých si myslia, že sú dôležité pre postup v projekte, ale v skutočnosti sa ich práca nevyužije alebo dokonca môže projektu viac poškodiť.

Dať ľuďom nástroje na rozpoznanie, či je ich práca len márnym časom, je jednou z najväčších povinností manažéra. Vedúci by mal definovať kolektívny zmysel pre postup, ktorý by zahŕňoval príspevok každého ako postupovať. Mal by byť premenený na niečo „hmatateľné“, aby mohli ľudia podľa toho postupovať.

Ako definovať postup

Podľa [1] sa definovanie postupu skladá z dvoch častí. Prvou je plánovanie práce a navrhnutie cieľov, ktoré je potrebné dosiahnuť. Druhou, na ktorú sa častejšie zabúda, je denné riadenie tímového postupu. Je to práca vedúceho, aby pravidelne zisťoval od ľudí v tíme, ako ich práca postupuje k cieľu. Na každom stretnutí, diskusii by mal vedúci tlačiť, pripomínať a motivovať tím napred, hociaké ciele sú. Manažér má predsa najlepšiu prehľad nad celým projektom a mal by ho zdieľať, tak často ako môže.

Meranie práce

Aby sme vedeli lepšie sledovať postup v softvérovom projekte, musíme zaviesť určitú metriku na jeho vyjadrenie. Podľa [1] je viacero vecí, ktoré sa dajú merať pri projekte, a tak pomôcť ľuďom zamerať svoje úsilie na niektoré veci. Existujú organizácie, kde sa merajú napríklad takéto veci:

- koľko hodín strávil zamestnanec v kancelárii,
- koľko napísal správ, špecifikácií,
- koľko odovzdal prezentácií,
- koľko mal telefónnych hovorov,
- koľko riadkov zdrojového kódu napísal,
- koľko chýb odstránil,
- koľko poslal e-mailov.

Tieto veličiny sa síce ľahko zmerajú, ale majú podľa mňa malú výpovednú hodnotu. Môžem opraviť viac chýb za určitý čas, tým budem podľa hodnotenia aktívnejší. No ja si môžem zámerne vyberať ľahšie chyby, ktoré opravím rýchlejšie.

No postupu v projekte tým veľmi nepomôžem. Rovnako ako aj počet riadkov kódu hovorí skôr o veľkosti kódu, ale nehovorí nič o jeho kvalite.

Existujú spôsoby ako merať postup, ktoré sú síce ťažšie, ale majú väčšiu hodnotu. Sú to otázky, ktoré nútia premýšľať a vedú k dialógu o práci. Sú to otázky typu:

- Ku ktorému cieľu prispela dnešná časť práce?
- Ako ďaleko nám pomohla priblížiť sa k tomu cieľu?
- Ako veľmi k tomu prispela (priblížila nás o 5%? Alebo o 50%?)?
- Je jej kvalita dostatočne vysoká?
- Ktoré ďalšie časti projektu sa touto prácou podporili alebo aktivovali?

Nájdenie správnej úrovne detailov pre ciele je zručnosť a dá sa získať až časom a skúsenosťami. Zvyčajne sa vytvorí hierarchia zdedených cieľov, pričom sa zužujú. Takto môžu byť niektoré pre celý projekt, niektoré pre tím a niektoré len pre jednotlivca.

Na meranie postupu jednotlivca môžu slúžiť otázky:

- Ako nás priblížila moja dnešná práca k celkovému cieľu?
- Splňa moja dnešná práca požiadavky na kvalitu?
- Koľko ľudí z môjho tímu som pozitívne ovplyvnil?
- Koľko prekážok som prekonal?
- Koľkým prekážkam som zamedzil vznik?
- Na čo zložité, ale základné som sa spýtal, čo ostatní nevedeli?
- Ako som nás zefektívnil v postupe k cieľu?
- Koľkým problémom som dnes čelil, ktoré nám bránia v rýchlejšom postupe?

Na meranie postupu celého tímu sa dajú použiť otázky:

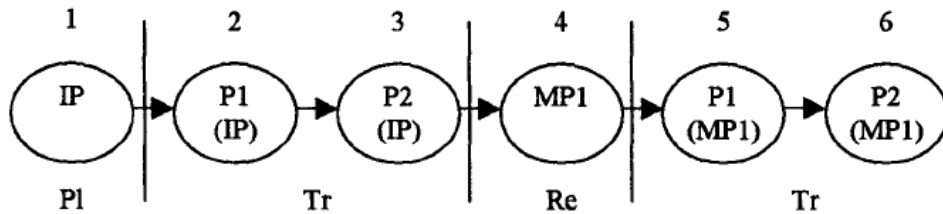
- Ako sme sa priblížili k naplneniu cieľa?
- Ako blízko sme k požadovanej úrovni kvality?
- Ktoré neadresované problémy nás spomaľujú?
- Je náš zostávajúci plán správny?

Sledovanie postupu

Podľa [2] pozostáva sledovanie postupu softvérového projektu z dvoch hlavných prístupov: periodickým a riadeným udalosťami. To znamená, že existuje viacero kontrolných bodov v čase (tzv. míľnikov), na ktorých je aktuálny stav postupu projektu viditeľný ako výsledok projektu. Každé takéto projektové plnenie korešponduje s každým kontrolným bodom. Postupnosť takýchto plnení tvorí zmenu histórie projektu. Projektová história zmien začína pôvodným plánom a končí kompletným projektom, ak sa nevyskytli žiadne vážne odchýlky od plánu.

Autori článku [2] vyvinuli nástroj na manažovanie a podporu integračných testov. Nazýva sa SMIT (z angličtiny Systém pre manažment jednotiek a integračných testov). Príklad histórie zmien je znázornený na Obr. 1. Začína pôvodným plánom (IP), keď je stav všetkých projektových aktivít a projektu ako takého len budúcnosť. Projekt potom postupuje podľa plánu. Ďalším kontrolným bodom (P1) sa zachytáva prvý stav podľa

pôvodného plánu. V nasledujúcom bode sa už stihlo vykonať viac aktivít (P2). Pôvodný plán je stále základom pre sledovanie postupu. Potom sa predpokladá, že bude potrebné urobiť opravné akcie. Povedzme, že výsledok poslednej dokončenej aktivity v treťom kontrolnom bode nás prinúti zmeniť počiatočný plán. Potom, ako ho zmeníme, sa zaznamená v kontrolnom bode 4 (MP1). Následne postupuje projekt podľa zmeneného plánu, ktorý slúži aj ako základ pre ďalšie sledovanie (P1 (MP1) a P2 (MP1)).



Obr. 1. Príklad histórie zmien projektu [2]

Prvý a štvrtý bod slúžia na počiatočný a modifikovaný plán a zdôrazňujú podstatu zmien, ktoré reprezentujú. Z Obr. 1. je zrejmé, že v histórii zmien sa nachádzajú dva hlavné typy zmien:

- veľké zmeny týkajúce sa plánu (body 1 a 4)
- menšie zmeny týkajúce sa sledovania plánu (body 2, 3, 5 a 6).

Sledovanie postupu projektu pozostáva z cyklu plánovania / preplánovania a sledovania. Na Obr. 1. je znázornený jeden takýto cyklus. Nástroj, o ktorom píše autori [2] umožňuje lepšie vytváranie histórie zmien. Má dve možnosti práce: plánovanie a sledovanie. Pri plánovaní sa definuje počiatočný plán. Pri sledovaní sa do histórie zmien pridávajú kontrolné body, ktoré vychádzajú z predchádzajúcich kontrolných bodov, pričom sú zaznamenané zmeny, ktoré nastali. Preplánovanie sa vyvolá pri sledovaní. Potom SMIT prenesie riadenie do plánovania, kde sa modifikuje pôvodný plán a zmení sa história zmien projektu pridaním nového kontrolného bodu.

Plánovanie pomocou SMIT

Pre plánovanie má SMIT štyri hlavné vlastnosti:

- prezentácia plánu úlohy veľkého softvérového projektu vo forme hierarchie sietí aktivít a hierarchie sietí objektov,
- podpora aktivít softvérového projektu a objektov odkazovaných v aktivitách,
- možnosť zahrnúť detaily plánovania jednotlivo a
- schopnosť automaticky generovať detailný plán jednotky.

Prvá z týchto vlastností je najpodstatnejšia pre históriu zmien.

Sledovanie pomocou SMIT

SMIT ponúka pre sledovanie projektu tri hlavné vlastnosti:

- podpora procesu tvorby histórie zmien,
- možnosť zahrňovať detaily sledovania jednotlivo a
- podpora sledovania jednotlivých jednotiek a integračných testov.

Najpodstatnejšia je opäť tá prvá. SMIT má prezentáciu postupu podobnú prezentácii plánu. Obsahuje hierarchiu sietí aktivít a sietí objektov a ich atribútov, ktorých je ale viac ako v prezentácii plánu. Atribút spomenutý v prezentácii plánu zmení zmysel na určitú základnú líniu v prezentácii postupu. Postup projektu sa tak meria v základných líniách počas sledovania.

Atribúty v prezentácii postupu nám umožňujú vytvárať časti histórie zmien bez preplánovania. Takéto časti sú napr. 1-3 a 4-6 na Obr. 1. Na vytvorenie časti 3-4 bolo za potreby vytvoriť modifikáciu plánu – preplánovanie. Modifikácia plánu vo všeobecnosti znamená modifikáciu v hierarchii prezentácii plánu aj vo vertikálnom aj horizontálnom smere. Podľa kontrolných zásad sa každá zmena v práci na produkte, v našom prípade v postupe, uchováva v podobe zmien – deltách. Postupnosť zmien má za následok akúsi reťaz delt. Pri postupe k poslednej verzii sa postupuje od začiatku reťazcom zmien, ktoré sa postupne aplikujú. Keďže existujú veľké a malé zmeny v histórii zmien projektu a prezentácie plánu a postupu obsahujú projektovú hierarchiu a atribúty, definujú sa tak dva druhy zmien:

- delta hierarchie - Δ_h ,
- delta atribútov – Δ_a .

Veľké zmeny ako plánovanie a preplánovanie vyžaduje obidve delty, malé zmeny v histórii (len ďalší kontrolný bod) vyžadujú iba uchovanie zmeny atribútov. História zmien projektu znázornená na Obr. 1. by teda mohla byť prezentovaná ako:

$(\Delta_h \Delta_a)(\Delta_a)(\Delta_a)(\Delta_h \Delta_a)(\Delta_a)(\Delta_a)$

Systém SMIT umožňuje niekoľko základných požiadaviek na sledovanie:

- získanie stavu postupu,
- zaznamenávanie postupu,
- kontrolu a zoskupovanie zaznamenaného postupu,
- aktualizovanie,
- uloženie jednoduchého kontrolného bodu,
- porovnanie dvoch kontrolných bodov toho projektu,
- definovanie detailov pri sledovaní,
- zobrazenie histórie míľnikov,
- iniciovanie preplánovania.

Prvých päť vlastností sú potrebné k postupu bez preplánovania. Zaznamenávanie postupu zahŕňa vyplnenie hodnôt príslušných atribútov. Aktualizovanie nastaví postup podľa toho, koľko z aktuálnych úloh sa stihlo dokončiť. Uloženie kontrolného bodu sa môže uskutočniť až po úspešnej kontrole, zoskupení postupu a aktualizácii.

Porovnávanie kontrolných bodov projektu je dôležitá časť spracovania histórie zmien. Môžu sa vybrať hociktoré dva body v postupe, pričom sa následne zobrazia zmeny, ktoré nastali vo vývoji z jedného po druhý. Porovnávanie, kde je ako prvý bod

nastavený počiatočný bod histórie zmien, sa nazýva globálne, všetky ostatné sa nazývajú lokálne. Globálne sa používa, keď sa porovnáva aktuálny stav postupu oproti pôvodnému plánu.

Porovnávanie využíva prezentáciu plánovania postupu. Môže mať väčšiu zrnitosť alebo menšiu. Pri väčšej sa porovnávajú iba zmeny v hierarchii postupu, teda zmeny v úrovniach a objektov. Pri menšej zrnitosti sa porovnávajú aj siete aktivít.

Manažér si môže prezerat' postup na rôznej úrovni detailov. Je to analogické ako pri naplánovaní detailov pri procese plánovania, ktoré sú všetky rovnocenné a na sebe nezávislé. Detaily pri sledovaní sú viazané na detaily pri plánovaní. Okrem toho všetky detaily pri sledovaní nemajú rovnakú dôležitosť. Stav atribútov má vždy najväčšiu prioritu.

Preplánovanie

Cyklus preplánovania začína ešte počas sledovania. Úloha získania postupu je prvá, ktorá sa musí splniť pri sledovaní. Ďalej musí nasledovať iniciovanie preplánovania. Má dva kroky. V prvom kroku zistí SMIT, či boli splnené podmienky na preplánovanie. Preplánovanie počiatočného bodu je možné len vtedy, ak je aspoň jedna elementárna aktivita, ktorá má stav dokončená. Druhým krokom je prenos riadenia plánovaniu a zaznamenanie budúcich aktivít do ďalšieho kroku oproti pôvodnému stavu do prechodného kroku. Nasleduje proces plánovania, ktorý vytvorí ďalší kontrolný bod z prechodného, kde zahrnie pôvodné hierarchie a atribúty. Následne sa kontrolný bod uloží do histórie.

Výhodou systému SMIT môže byť podľa môjho názoru práve podpora zmien v projekte. Môže to umožňovať lepší pohľad na vzťahy medzi plánovaním a sledovaním projektu. Môže to byť práve výhoda oproti iným nástrojom, ktoré sú dostupné v súčasnosti, ktoré vedia porovnať dva stavy v postupe z pohľadu sledovania. Ale nemajú možnosť preplánovať projekt. Ďalej môže pomôcť manažérovi v tom, že ho navedie, keď nastane potreba preplánovať projekt, ako aj navrhne, ktoré časti projektu treba v pláne modifikovať.

V porovnaní s Microsoft Project vyšiel SMIT úspešnejšie v tom, ako odhadol počet iterácií na projekt. Obidvom nástrojom bol zadaný rovnaký počet aktivít n , ktoré boli medzi sebou rôzne prepájané – sériovo alebo paralelne. V tabuľke Tab. 1. je vidieť, že redukcia počtu iterácií bola použitím SMIT-u nižšia o 61%-88% .

n	Počet iterácií		Redukcia v %
	MS Project	SMIT	
6	51	20	61
30	330	44	87
60	600	74	88

Tab. 1. Porovnanie počtu iterácií medzi MS Project a SMIT [2]

Záver

Ak sa raz ocitneme v pozícii manažéra, budeme sa musieť postarať o nevyhnutné zmeny, ktoré sa počas softvérového životného cyklu vyskytnú. Určite sa nezaobídeme bez plánu. Podľa plánu môžeme potom sledovať postup v softvérovom projekte. V prípade rozdielov oproti pôvodnému plánu môže manažér vykonať príslušné opatrenia ako nasmerovať ľudí, určiť, ktoré úlohy sú dôležitejšie a v konečnom dôsledku preplánovať úlohy v projekte.

Na to dnes slúži už veľa nástrojov. Niektoré vedia stav projektu posúdiť objektívnejšie iné nie, niektoré sú výhodnejšie pri kontakte so zákazníkom. Manažér by mal ale hlavne komunikovať s celým tímom, povzbudzovať a pomôcť riešiť problémy, hoci aj keď nevie presne, ako ich vyriešiť.

Použitá literatúra

1. Berkun, S: *Work vs. Progress*, August 2005
<http://www.scottberkun.com/essays/essay45.htm> dňa 20.10.2006
2. Mihajlovic, Z. Velasevic, D.: *Tracking Software Projects with the Integrated Version Control in SMIT*. Software Engineering Notes vol. 26, Marec 2001

Annotation

Tracking software project and influence on planning and control

During development software project management has to find the way how to track its progress. Solve problems with performance, planning, costs and control on the way to successfully finishing the project. It is nothing extraordinary when the project's time extends or the costs increase. Good management task is to establish automatic project tracking. Management should make and establish system to track project progress and control plan. From acquired values determine status and adjust the developing process concerning planning and control. This automatic tracking can alert managers on incoming problems and allows them to take action. This paper contains some of the techniques used by tracking the software project progress.