

Dôveruj, ale preveruj (Monitorovanie v softvérovom inžinierstve)

MARIAN MAJAN

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
majan04@student.fiit.stuba.sk*

Abstrakt. Je všeobecne známe, že na softvérovom projekte sa vo väčšine prípadov pracuje niekoľkokrát dlhšie, ako bolo pôvodne naplánované. Preto dôležitou súčasťou manažmentu softvérového projektu je nielen kvalitné plánovanie ale aj monitorovanie stavu, v akom sa projekt nachádza. Manažér je potom schopný včas identifikovať rozdiely voči pôvodnému plánu a následne správne rozložiť zdroje na jednotlivé etapy projektu. Tým sa zvyšuje možnosť eliminácie prílišného predraženia výsledného produktu či neskorého ukončenia jeho vývoja. Monitorovanie softvérového projektu však zároveň predstavuje zložitý proces, na ktorom sa musia kooperatívne zúčastňovať tak manažéri ako aj programátori. Cieľom tejto eseje je poskytnúť pohľad autora na problematiku monitorovania softvérového projektu v rôznych aspektoch. Zaoberá sa dôvodmi potreby monitorovania softvérového projektu, vyberá jeho hlavné výhody a nevýhody. Rozoberá príčiny narušenia plánovaného priebehu projektu. Nakoniec stručne opisuje prostriedky, ktoré pomáhajú monitorovať projekt počas doby jeho vývoja.

Úvod

Aj napriek tomu, že softvérové inžinierstvo má už pomaly 40 rokov, ešte stále sa jeho princípy nestali pevnou súčasťou vývoja softvérových produktov. Mnohé zo spoločností, ktoré sa zaoberajú vývojom nových programov akosi zabúdajú na potrebu manažmentu. Potom sa bežne stáva, že termín ukončenia práce na projekte sa nestihne a projekt sa niekoľkokrát predraží. Či sa nám to páči alebo nie, faktom ostáva, že len v málo firmách tvorí manažment každodennú súčasť vývoja.

Dôvodov, prečo je to tak, by sme zrejme našli niekoľko. Jednak sú to zažité spôsoby programovania „na kolene“, kedy sa veci ako plánovanie, monitorovanie zdajú byť zbytočným luxusom, ktorý nám pre náš projekt aj tak netreba. Opak je však vo väčšine prípadov pravdou. Častokrát vo vývojom tíme chýba kvalifikovaná osoba, ktorá by bola schopná uviesť všetky princípy manažmentu v softvérovom inžinierstve

do praxe. Stále treba mať na pamäti, že manažovanie softvérového projektu nie je jednoduchá činnosť a vyžaduje si nemalé znalosti a skúsenosti.

Titul tejto eseje - „Dôveruj, ale preveruj“ nebol zvolený náhodne. Nemyslím len dôveru k podriadeným - členom tímu a overovanie ich pracovného nasadenia, výsledkov, ale dôveru chápem v širšom slova zmysle, ako dôveru k pôvodnému plánu. Nejednen manažér má dojem, že kvalitný plán je postačujúcou požiadavkou pre správne manažovanie projektu. O to nepríjemnejšie je však vytriezvenie, keď zistí, že dokonalý plán ostal len plánom a v skutočnosti sa projekt nachádza niekde v prvých fázach vývoja. Náprava tohto stavu si vyžaduje nemálo financií a času. Preto úlohou dobrého manažéra nie je len zostrojenie kvalitného plánu ale aj monitorovanie stavu, v akom sa projekt aktuálne nachádza a porovnanie tohto stavu s naplánovaným.

Potreba monitorovania softvérového projektu

Základným kameňom softvérového projektu je bezpochyby plán. Aby však na ňom mohol manažér stavať, musí byť vytvorený s veľkou precíznosťou. Je potrebné, aby sa na jeho výrobe podieľali všetci členovia vývojového tímu – od analytikov, návrhárov, programátorov, ľudí zodpovedných za testovanie až po manažérov. Len dobrou komunikáciou v rámci tímu je možné vytvoriť plán, ktorý je reálne splniteľný. Musia byť do neho zapracované všetky požiadavky členov tímu, zároveň však musí vychádzať z empirickej skúsenosti manažéra. Podľa môjho názoru skúsený manažér, ktorý si sám „odžil“ aj iné, nemanadžérske role, napr. pôsobil ako programátor či analytik, má veľmi dobrý nadhľad na jednotlivé požiadavky členov tímu. Vychádzajúc z vlastnej skúsenosti vie odhadnúť reálnosť jednotlivých požiadaviek, identifikovať kritické miesta v pláne a pridať pre ne dostatočnú časovú rezervu, dokáže správne rozvrhnúť zdroje.

Ciele plánovania pritom nie sú len vytvoriť určitú štruktúru postupu na projekte ale aj [1]:

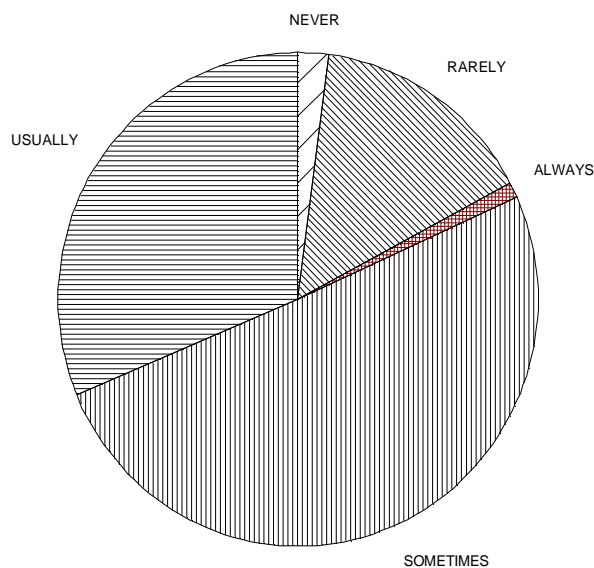
- redukovať neurčitost' výsledku projektu,
- zlepšiť výkonnosť,
- zverejniť ciele projektu a tým zabezpečiť ich lepšie porozumenie,
- vytvoriť základňu pre sledovanie a riadenie práce.s

Ako som však už spomenul vyššie, plán je sám o sebe nepostačujúci. Projekt, ktorý síce má vytvorený veľmi dobrý plán avšak absentuje u neho spätná odozva reálneho stavu projektu, bude len málokedy zrealizovaný podľa predstáv manažéra. Potom úspešnosť množstva projektov je nepostačujúca, zvyšuje sa množstvo času potrebného na vývoj, zvyšuje sa aj cena a nakoniec mnohokrát sa znižuje kvalita. V [2] je uvedený prieskum, z ktorého jasne vyplýva, že pri veľkom množstve softvérových projektoch sú prekročené tak termíny ako aj náklady. Až pri 40% všetkých sledovaných projektov došlo k nedodržaniu plánovaných termínov. Ďalej je tu uvedený prieskum, v ktorom korešpondenti odpovedali na otázky ohľadom prekročenia termínov a výslednej ceny produktu. Až 82% z opýtaných uviedlo, že pri ich

projektoch došlo k omeškaniu (graf je zobrazený na Obr. 1). Naopak, iba 16% korešpondentov uviedlo, že pri realizácii projektov nedošlo k prekročeniu nákladov (alebo k nemu došlo len zriedka).

Pri mnohých z týchto projektov nastal problém v práve zle vytvorenom pláne a/alebo podcenením monitorovania skutočného stavu projektu voči plánu. Chyby, ktoré postupne v rámci vývoja vznikali neboli včas odhalené a prenášali sa ďalej, hlbšie do projektu, kde postupne spôsobovali väčšie a väčšie problémy. Nakoniec sa snaha stihnúť všetko podľa plánu ukáže ako neuskutočniteľná. Pokiaľ je projekt v neskorom štádiu riešenia, nepomôže už ani rozšírenie zdrojov – zapojenie viac ľudí do projektu. Existuje všeobecne uznávaný Brooksov zákon ktorý hovorí, že pridanie inžinierov do oneskoreného projektu ho oneskorí ešte viac [2]. Neskoré odovzdanie projektu však v komerčnej sfére mnohokrát znamená jeho neakceptáciu a teda vedie k ukončeniu celého projektu. Všetky prostriedky do neho vložené boli vynaložené zbytočne. Pri rozsiahlych projektoch sa tak jedná o naozaj veľké sumy, ktoré sú „pustené dolu vodou“.

Vidíme, že ak projekt nechceme ešte pred začatím vývoja odsúdiť na neúspech, ak chceme vytvoriť naozaj úspešný projekt, v ktorom obmedzíme riziko prekročenia termínov a nákladov a bude zachovaná aj jeho požadovaná kvalita, musíme okrem dobre pripraveného plánu zabezpečiť aj monitorovanie aktuálneho stavu v akom sa projekt nachádza. Monitorovanie stavu projektu nám umožní odhaliť kritické miesta a alokovať pre ne dostatočný počet zdrojov. Tiež sú automaticky odhalené najmenej kritické miesta, pri ktorých je možné uvoľniť prostriedky.



Obr. 1: Neskoré odovzdanie projektu [2].

Monitorovanie softvérového projektu je však dôležité aj v rámci spätnej väzby medzi členmi tímu a manažérom. Väčšina ľudí má tendenciu nechávať si všetko na poslednú chvíľu a na projekte začať intenzívne pracovať „o 5 minút 12“. Pokiaľ jediným hraničným termínom, jediným kontrolným bodom je ukončenie projektu a finálne odovzdanie produktu, môže táto „výsada“ členov tímu znamenať pre projekt jeho neúspešný zánik.

Nakoniec, monitorovanie softvérového projektu tvorí aj nástroj pre komunikáciu medzi zákazníkom a manažérom. Motivácia manažéra nemusí byť dostatočná, pokiaľ v projekte nie je priamo (najmä finančne) zainteresovaný, prípadne mu nehrozia žiadne sankcie za nedodržanie termínov a prečerpanie zdrojov. Preto je potrebná aj spätná kontrola manažéra zákazníkom, najmä ak zákazník už v počiatočnej fáze do projektu investoval. Manažér sa stáva komunikačným kanálom, ktorý od členov tímu získava aktuálny stav projektu a prezentuje ho zákazníkovi. Zákazník je tlačný vlastným kapitálom, manažér zákazníkom a členovia tímu manažérom. Všetko riadi plán a aktuálny stav získaný monitorovaním.

Monitorovanie postupu

Ako už bolo spomenuté vyššie, základom úspešného softvérového projektu je nielen dobrý plán, ale aj monitorovanie stavu v akom sa projekt voči plánu nachádza.

Monitorovanie postupu v softvérovom projekte je pri tom veľmi zložitá úloha, ktorá si vyžaduje dobrú (a pravdivú) komunikáciu medzi manažérom a ostatnými členmi vývojového tímu. Úlohy monitorovania sú podľa [4]:

- sledovať a konfrontovať aktuálny pokrok projektu a výsledky s plánom projektu,
- korigovať plán projektu tak, aby odrážal aktuálny stav projektu,
- poskytnúť pohľad na postup v projekte, takže tím ako aj manažment vie včas vykonať akcie v prípade, že projekt zaostáva oproti pôvodnému plánu.

Úlohou manažéra v tomto procese je neustále dotazovanie všetkých podriadených a zisťovanie stavu úloh na ktorých pracuje. Aby však projekt bol cez jednotlivé úlohy sledovateľný, vyžaduje si, aby bol ešte v štádiu plánovania rozdelený na dostatočné množstvo úloh. Tiež musí byť jasne určený termín do kedy má byť úloha splnená a vymedzené zdroje. Potom manažér vidí, s akou úspešnosťou sú riešené jednotlivé úlohy.

Takto nastavené monitorovanie projektu ľahko identifikuje kritické úlohy v projekte. Manažér je schopný upraviť plán, vyhradiť pre tieto úlohy viac zdrojov. Zároveň (samozrejme, za predpokladu, že bol vytvorený korektný plán) dostáva spätnú väzbu na výkonnosť jednotlivých členov tímu, ktorú vie neskôr zohľadniť pri ich ocenení, či príprave plánu pre ďalší projekt.

Problémy monitorovania

Monitorovanie je zložitý proces, vyžaduje si zodpovedný prístup od tímu aj od manažmentu. Práve akási jeho subjektivita a závislosť na členoch tímu je jeho slabinou.

Pre väčšinu programátorov predstavujú veci ako plánovanie či monitorovanie nočnú moru. Sú zvyknutí na slobodný prístup k programovaniu, kedy nikým nie sú obmedzovaní a „nezaťažujú“ sa vecami ako sú termíny či zdroje. Už z povahy procesu programovania softvéru je určenie potrebných zdrojov pre mnohých z nás, programátorov, neľahkým orieškom.

Navyššie ak dôjde prvotnému podceneniu úlohy programátorom (či iným členom tímu) ešte v štádiu plánovania a manažér to nebol schopný odhaliť, nastáva problém. Väčšina ľudí si nerada priznáva svoje chyby. V štádiu monitorovania následne zlyhá komunikácia medzi manažérom a členmi tímu na neochote priznať si svoju chybu. Komunikácia zo strany členov tímu je zavádzajúca (klamlivá). Ak tento stav manažér nepostrehne, proces monitorovania väčšinou zlyhá a s ním zväčša aj celý projekt.

Ďalším problémom je relatívny pohľad na splnenie jednotlivých úloh. Preto ak úlohy nie sú dostatočne malé, môže nastať prehnaný pocit bezpečia, kedy člen tímu bude manažérovi tvrdiť (a sám o tom bude aj presvedčený), že úloha je splnená na 90%. V takomto stave však úloha častokrát stagnuje a práca na zvyšných desiatich percentách trvá niekoľkonásobne dlhšie.

Samozrejme, ako už bolo spomenuté, skúsený a dobre pripravený manažér je schopný odhaliť väčšinu z načrtnutých problémov v ich rannom štádiu a zamedziť rozšíreniu chyby a tým aj ohrozeniu celého projektu.

Identifikovali sme teda hlavné problémy monitorovania softvérového projektu, a to *závislosť na pôvodnom pláne*, ako aj *závislosť na korektnom prístupe zo strany tímu* a *závislosť na kvalitách manažéra*.

Pomocné prostriedky

Existujú rôzne prostriedky, ktoré nám umožňujú sledovať postup práce na projekte. Vzhľadom na to, že esej sa zaoberá monitorovaním softvérového projektu, bolo by veľmi krátkozraké úplne odignorovať problém metriky či dostupných softvérových prostriedkov.

Metrika nám poskytuje pohľad na projekt prostredníctvom čísiel. Dnes je k dispozícii veľké množstvo rôznych metrík, niektoré majú vysokú výpovednú hodnotu, niektoré zase len málo vypovedajú o skutočnom pokroku. Preto je potrebné pre každý projekt starostlivo vyberať správnu metriku. Vzhľadom na povahu tejto práce nie je cieľom podrobne oboznámiť čitateľa s rôznymi metrikami v monitorovaní softvérového projektu. V prípade záujmu sa môže obrátiť napríklad na [3].

Čo sa týka dostupných softvérových prostriedkov, ktoré sú manažérovi počas manažmentu projektu k dispozícii, tých už dnes existuje veľké množstvo. Sú nápomocné pri príprave plánu, vedia odhaliť kritickú cestu. Tiež zlepšujú komunikáciu v rámci tímu, poskytujú aktuálnu verziu projektu pre všetkých členov.

Existujú však aj prostriedky, ktoré nám umožňujú zlepšiť sledovanie prác na projekte a zobrazenie dosiahnutého pokroku. Okrem iných môžeme spomenúť Ganttovu schému. Tento podporný prostriedok je veľmi užitočný najmä pri plánovaní ale aj pri monitorovaní softvérového projektu. Umožňuje rozložiť jednotlivé úlohy na podúlohy, stanoviť termíny, kedy majú byť splnené. Tiež umožňuje stanoviť zdroje medzi jednotlivé úlohy. Po začatí práce na projekte je možné zadávať úspešnosť vyriešenia jednotlivých úloh. Veľmi prehľadne zobrazuje pokrok, ktorý bol dosiahnutý na úlohách, je možné rýchlo identifikovať úlohy, ktoré zaostávajú, alebo naopak, úlohy ktoré boli ukončené s predstihom.

Samozrejme, existujú aj špecializované softvérové produkty, ktoré zohľadňujú jednotlivé vybrané metriky sledovania pokroku projektu.

Záver

Každodenne sa stretávame s potrebou vykonávať určité činnosti. Vytvárame si plán, ako budeme na týchto činnostiach pracovať, do kedy ich dokončíme. Zároveň však väčšina z nás stanovený plán prekročí. O to kritickejšia je situácia pri manažmente softvérového projektu, ktorý sa komplikovanosťou nedá so spomenutými úlohami porovnať.

Monitorovanie predstavuje veľmi dôležitú etapu manažmentu softvérového projektu. Manažérovi poskytuje pohľad na reálny stav, v akom sa projekt nachádza, znižuje celkové riziko prekročenia termínov či prečerpania prostriedkov.

Monitorovanie môže v určitých špecifických prípadoch jeho nezvládnutia podať o stave riešenia projektu klamlivé, príliš optimistické informácie. Napriek tomu by ale monitorovanie malo byť stabilnou súčasťou každého softvérového projektu, pretože zaradenie monitorovania do projektu vysoko zvyšuje šancu projektu na úspech.

Použitá literatúra

1. Bieliková, M.: *Manažment v softvérovom inžinierstve*. Bratislava. 1999.
2. Genuchten, M.: Why is Software Late? An Empirical Study of Reasons for Delay in Software Development. In: *IEEE Transactions on Software Engineering*, 1991, 582-590.
3. Linden, J.: *Monitoring Progress in Software Development*. 2003. <http://www.stsc.hill.af.mil/crosstalk/2003/07/vanderLinden.html> (25. 10. 2007).
4. *Process for Project Monitoring and Control*. 2003. <http://www.dir.state.tx.us/eod/qa/monitor/index.htm> (25. 10. 2007).

Annotation

Trust but verify (Software project monitoring)

It is general knowledge, that work on many software project is several time overrun. To avoid this, it is very important to make high quality project plan and also to monitor project work progress. Manager is than able to identify how project varies form original plan and divide project's sources to project's tasks. This process can avoid cost and time overrun. Software project monitoring is difficult process, which require cooperation between managers and programmers. The goal of this essay is to offer author's point of view at software project monitoring in several aspects. Author describes reasons why to monitor software project, and he identifies it's advantages and disadvantages. Next he presents reasons, that cause project failure. The end of the paper contains analysis of some tools, that are used to monitor project.