

MONITOROVANIE SOFTVÉROVÉHO PROJEKTU - CIELE A POSTUPY

Práve vďaka úspešnému zvládnutiu monitorovania projektu sa môže vysnívaný cieľ stať skutočnosťou.

Miloš Blaško

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

Abstrakt. Monitorovanie stavu a vývoja projektu je dôležitou súčasťou manažmentu softvérových projektov, bez ktorého sa dodržanie dohodnutých termínov či plánovaného rozpočtu stáva iba zbožným želaním členov tímu. Od chvíle, kedy sa do rúk manažérov dostane plán vývoja softvérového projektu, ich prvoradou úlohou sa stáva sledovanie a monitorovanie prác zúčastnených tímov. Cieľom tejto náročnej činnosti je včas identifikovať vzniknutý problém v určitej fáze projektu a podniknúť adekvátne kroky k minimalizovaniu prípadných škôd. V mojej eseji sa venujem zhodnoteniu obmedzení, ktoré nepriaznivo vplyvajú na relevantnosť a precíznosť plánovania, ktoré je podkladom pre následné monitorovanie projektu. Ďalej sa zameriavam na zhodnotenie cieľov a postupov samotného procesu monitorovania spolu s identifikovaním príčin vzniknutých anomálií a spôsobu ich nápravy. Záver je venovaný možným následkom po zle zhodnotených zásahoch do vývoja softvérových systémov.

Kľúčové slová: obmedzenia pri plánovaní, proces monitorovania, príčiny anomálií, stabilizácia vývoja, nevhodný zásah

Úvod

Každý softvérový projekt sa začína vytvorením požiadaviek, ktoré musia v minimálnej forme zodpovedať nasledujúce tri základné otázky: Čo? Dokedy? Za koľko? Úlohou

analytikov je vytvorenie plánu, podľa ktorého sa bude projekt uberať po svojej kľukatej ceste až do cieľa. Do akej miery bude jeho cesta náročná a to, či pretrhne pomyselnú cieľovú čiaru s úsmevom na tvári alebo so slzami v očiach, závisí teraz okrem iného aj na manažéroch. Tí po dôkladnom preštudovaní práce kolegov analytikov začnú s rozdeľovaním úloh medzi jednotlivé vývojárske tímy. Hneď ako sa začnú práce na tvorbe systému, novou úlohou manažérov sa medzi inými stáva práve monitorovanie bežiacich procesov vývoja.

Zmyslom analýzy napredovania kontrolovaných častí projektu vzhľadom k určenému plánu je včasné odhalenie problémových procesov. Ak k nim manažéri nezaujmú primeraný postoj a neidentifikujú príčinu ich vychýlenia, projektu hrozí nedodržanie časových alebo aj rozpočtových hraníc, v tých najhorších scenároch dokonca jeho zánik. Cieľom monitorovania sa teda stáva minimalizovanie konečného rozdielu medzi použitými a plánovanými zdrojmi prostredníctvom včasnej identifikácie anomálií v procese vývoja a ich adekvátneho riešenia.

V prvej kapitole tejto eseje sa venujem stručnému popisu obmedzení, ktorým musia analytici čeliť pri tvorbe plánov vývoja softvérových projektov. Za ňou nasleduje kapitola venovaná cieľom a postupom procesu monitorovania. Obsahom tretej kapitoly je určovanie príčin vychýlenia kontrolovaných procesov, pričom vo štvrtej kapitole uvádzam príklady zásahov napravujúcich vzniknuté problémy. V piatej kapitole stručne rozoberám následky nesprávnych zásahov do procesu vývoja softvérových projektov.

Prečo sa takmer vždy na niečo nemyslelo?

Odpovedí na túto otázku je samozrejme viacero. Čím viac ich zhromaždíme a preskúmame, tým viac sa zvyšuje pravdepodobnosť, že si s odhalenými obmedzeniami dokážeme v budúcnosti poradiť. Teraz sa pokúsim oboznámiť čitateľa s najvýznamnejšími problémami, s ktorými sa analytici stretávajú pri ich práci.

Aby sme mohli navrhnúť riešenie problému a úspešne ho realizovať, vyžaduje sa od nás istá úroveň vedomostí a skúseností z problémovej oblasti [44]. V dnešnej dobe sa ale stále populárnejšou stáva špecializácia (či už širšia alebo užšia) ako firiem, tak aj samotných zamestnancov. Keď sa vedenie rozhodne pre realizáciu zákazky z domény, s ktorou nemali vývojové tímy doteraz žiadne skúsenosti, je vysoko pravdepodobné, že odhady zahrnuté v plánovaní nebudú dostatočne presné a kompletne.

Za podobnú situáciu môžeme považovať prípad, keď má projektový tím za úlohu vytvoriť úplne nový typ systému, ktorý nemá žiadneho predchodcu. Keďže nie sú k dispozícii žiadni odborníci na danú tému, jediné, o čo sa môže tím oprieť, sú predstavy a požiadavky zákazníkov [44]. Na jednej strane by mali byť zadávatelia zákazky dostatočne zbehlí v problémovej oblasti, ale na strane druhej od nich s určitosťou nemôžeme očakávať znalosti z tvorby softvérových projektov. Z vlastnej skúsenosti viem, že cieľoví užívatelia majú veľmi bohatú predstavivosť a na nešťastie vývojárov veria, že naprogramovať sa dá všetko. Aj keby bolo možné naprogramovať naozaj všetko, úlohou analytika je dôkladne zvážiť realizateľnosť a vhodnosť návrhu vyplývajúceho z ich požiadaviek. Taktiež odhad koncových termínov a rozpočtu je v tomto prípade veľmi náročný a môže viesť k neuskutočniteľným dohodám so zákazníkom.

Okrem problémov pri návrhu vyvíjaného systému musíme počítať aj so zmenami požiadaviek počas prác na projekte. Každá zmena vlastností systému si vyžaduje prehodnotenie a prípadné doplnenie aktuálnych plánov.

Keďže uvedené nedostatky a prekážky v procese plánovania sú v praxi bežné, manažéri sa nemôžu riadiť striktne podľa prvotného plánovania a tomuto faktu by mali prispôbiť aj proces monitorovania softvérového projektu.

Proces monitorovania – ciele a postupy

Úlohou manažéra zodpovedného za monitorovanie softvérového projektu, teda samotným cieľom monitorovania nie je splnenie pôvodného plánu do bodky, ale minimalizovanie rozdielu medzi naplánovanými a použitými zdrojmi. Všetky procesy prebiehajúce v rámci monitorovania by mali viesť k odovzdaniu hotového produktu v takej podobe, s akou budú všetci zúčastnení pracovníci spolu s vedením stotožnení a spokojní. Práve vďaka úspešnému zvládnutiu monitorovania projektu sa môže vysnívaný cieľ stať skutočnosťou. Niet pochýb o tom, že samotné monitorovanie nemôže spasiť zle navrhnutý alebo naplánovaný projekt. Avšak v prípade, ak boli tieto časti svedomito vypracované, má zmysel vložiť zostávajúce sily manažéra práve do procesu monitorovania.

Po predstavení cieľov monitorovania sa dostávam k samotným postupom, ktoré sú pri monitorovaní najviac využívané. Pre potrebu zjednodušenia celkovej organizácie sa projekt ako taký rozdelí na menšie procesy alebo podúlohy, ktoré sú následne pridelené vývojovým tímom. Až do ukončenia všetkých prác na vývoji softvéru prebieha v rámci monitorovania neustále porovnávanie súčasného a predpokladaného stavu určeného plánom projektu. Tento proces zahŕňa meranie kvality zvolených častí vývoja v čase a identifikovanie anomálií v získaných údajoch. Medzi časté výchyly patrí napríklad zníženie produktivity, nárast chybovosti alebo neprimeraná doba realizácie (príliš krátka alebo dlhá) [11]. V prípade zaregistrovania odchýlky je nutné podniknúť nápravné kroky (zasiahnuť do procesu vývoja), ktoré by mali minimalizovať jej dopad na celkový priebeh projektu.

Anomálie možno vo všeobecnosti považovať za akési šumy objavujúce sa v procese vývoja systému, následky neznámych vplyvov (pozitívne aj negatívne), ktoré v konečnom dôsledku znižujú úroveň predvídateľnosti budúceho vývoja.

Medzi možné vplyvy, ktoré pôsobia na plánovaný chod projektu patria napríklad [11]:

- predstavenie nového CASE prostriedku
- použitie novej testovacej alebo inšpekčnej techniky, ktorá znižuje výskyt chýb odhalených až po odovzdaní softvéru
- vplyv nedostatočnej dokumentácie na budúcu údržbu softvéru
- prítomnosť neznámej chyby v používanom hardvéri alebo operačnom systéme
- nepredvídaná neprítomnosť kľúčovej osoby

Pri sledovaní a ohodnocovaní spomínaných procesov majú manažéri k dispozícii rôznorodé nástroje, ktoré zefektívňujú vykonávanie mnohých úkonov. Tieto nástroje slúžia ako indikátory stavov, v ktorých sa procesy nachádzajú. Ak sa niektorý z nich

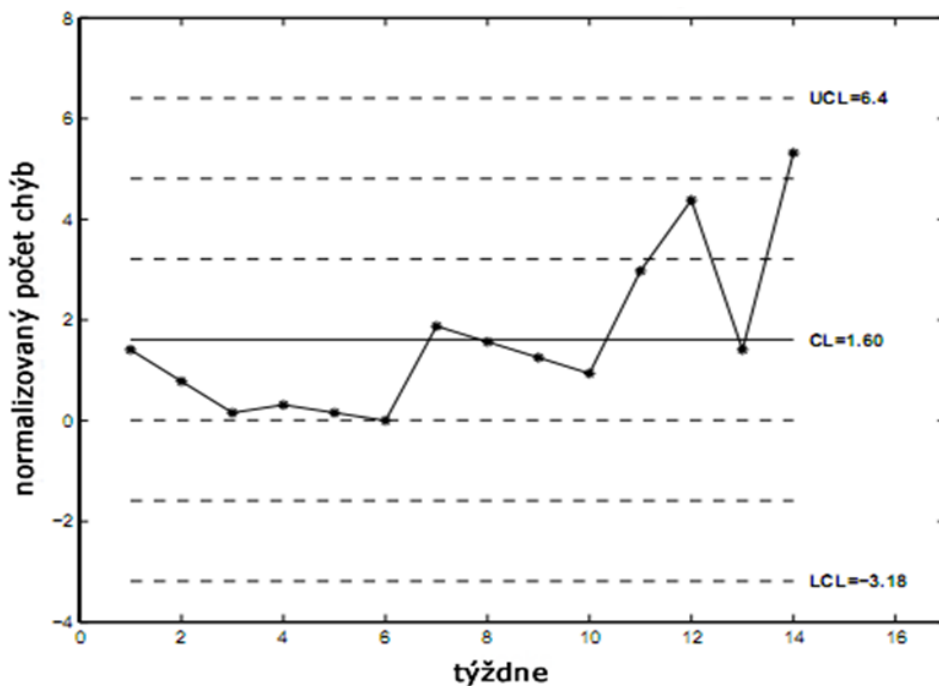
dostane do stavu, ktorý sa líši od predpokladaného, nástroj oboznámi manažéra o konkrétnych nezrovnalostiach. Ten na základe analýzy obdržaných dát dokáže identifikovať príčinu vzniku narušenia procesu. V prípade zaregistrovania odchýlky je nutné podniknúť nápravné kroky (zasiahnuť do procesu vývoja), ktoré by mali minimalizovať jej dopad na celkový priebeh vývoja.

K najbežnejším indikátorom patrí metóda využívajúca míľniky (z angl. milestones), teda konkrétne termíny, ktoré musia jednotlivé paralelne pracujúce tímy splniť. Pre každú fázu vymedzenú práve spomínanými míľnikmi sa odhadne úsilie potrebné na jej úspešné ukončenie [44]. Samotné monitorovanie je potom založené na pravidelných stretnutiach manažéra s celými tímami alebo aspoň s ich vedúcimi. Na týchto stretnutiach manažér zhromažďuje informácie o aktuálnom stave prác na projekte. Skombinovaním dát získaných pri stretnutiach a dát vyprodukovaných pomocnými nástrojmi vzniká základ pre následnú analýzu vzniknutej situácie.

Medzi ďalšie metódy monitorovania vývoja softvéru patrí štatistické sledovanie výskytu chýb, prípadne sledovanie množstva investovaných výdavkov. V nasledujúcej časti sa pokúsím vyjadriť práve k štatistickému sledovaniu výskytu chýb.

Kontrolovaný proces sa nachádza v stabilnom stave dovtedy, pokiaľ sa počet identifikovaných chýb pohybuje v stanovenom rozmedzí. Určenie rozmedzia si vyžaduje znalosť náročnosti spracováanej fázy projektu ako aj úrovne schopností tímu, ktorému bude práca pridelená. Závisí aj od dôležitosti monitorovanej časti systému. Pri kritických fázach by som zvolil menšie rozmedzie, pretože aj výchyľky menšieho charakteru môžu byť potencionálne nebezpečné. Naopak pre okrajové procesy alebo procesy, pri ktorých sa s výchyľkami počíta, by som zachoval väčšie rozmedzie, ktoré pri signifikantných výchyľkách splní svoj účel, ale nebude proces monitorovania zbytočne zaťažovať.

Príklad výsledkov takéhoto sledovania je znázornený na Obr.1.



Obr.1. Príklad diagramu kontroly (z angl. control chart) [33].

Hraničné hodnoty z obrázka predstavujú:

- LCL – dolný kontrolný limit (z angl. lower control limit)
- CL – centrálna línia (z angl. central line)
- UCL – horný kontrolný limit (z angl. upper control limit)

Čo spôsobilo vychýlenie procesu od plánu?

Čo nám môže toto zobrazenie povedať o stave skúmaného procesu? Vezmem do úvahy prípad, v ktorom sa väčšina hodnôt nachádza pod centrálnou líniou ako je to do ôsmeho týždňa na Obr.1. Skúmaný stav naznačuje, že priemerný počet odhalených chýb je menší ako sa predpokladalo. Podľa môjho názoru to môže byť buď dôsledkom zanedbania kontroly a testovania alebo v tom lepšom prípade dôsledkom prílišného pesimizmu pri určovaní kontrolných limitov. Ako teraz určiť, čo bolo skutočnou príčinou vychýlenia? Najlepšou cestou by mala byť priama komunikácia s členmi tímu, ktorí sú s danou odchýlkou spojení. Po zozbieraní všetkých dostupných informácií môže manažér začať s ich podrobnou analýzou. Pri jej vytváraní sa môže oprieť o svoje osobné skúsenosti alebo požiadať o doplnujúci názor kolegu, ktorý lepšie pozná konkrétny tím.

Príčin, ktoré mohli ovplyvniť môj modelový príklad, je niekoľko. Ak došlo k zanedbaniu testovania vytvorených častí softvéru, mohlo to byť spôsobené neprítomnosťou viacerých členov testovacieho tímu z dôvodu chrípkovej epidémie. Čo ak sa koreň problému nachádza v testoch samotných? V prvých týždňoch danej fázy je pochopiteľné, že dôraz pri testovaní nemusí byť kladený na kľúčové oblasti, pretože ich identifikovanie nemusí byť jednoznačné. Z pozorovaných výsledkov môže vyplynúť aj fakt, že skúmaná úloha, ktorou bol tím poverený, nebola primeraná jeho schopnostiam a jeho potenciál je nevyužitý.

Podobných príčin vychýlenia pozorovaného stavu by som mohol uviesť oveľa viac. Pre odstránenie každej z nich je potrebné podniknúť rôzne kroky a mnohé nás môžu priviesť k odhaleniu chýb v našich pôvodných plánoch. Toto bola názorná ukážka významnosti procesu monitorovania pri odhaľovaní anomálií vo vývoji softvérových systémov.

Máme vinníka. Čo s ním?

Po zaznamenaní odchýlky a následnom identifikovaní jej spúšťača je ďalšou úlohou manažéra podstúpiť adekvátne kroky na zaručenie stabilizácie daného procesu. Ešte pred vykonaním ľubovoľnej nápravnej akcie je na mieste položiť si otázku, či je vôbec potrebné reagovať na vzniknutú situáciu. Ak bol príčinou anomálie špeciálny prípad (z angl. Special cause) ako spomínaná chrípková epidémia alebo porucha dôležitej časti hardvéru, nie je nutné priamo zasahovať do bežiaceho procesu vývoja. V prípade, že bol vývoj pozastavený na dlhšiu dobu, čo by sa mohlo odzrkadliť pri dodržiavaní dohodnutých termínov, je nutné buď upraviť zodpovedajúce termíny, zvýšiť náklady alebo sa pokúsiť vzniknuté zameškanie dobehnúť.

Okrem zmien plánovaných termínov alebo rozpočtu má manažér k dispozícii aj iné spôsoby, akými môže stabilizovať vychýlené procesy. Za jeden z najefektívnejších dostupných prostriedkov považujem motiváciu ako pozitívnu tak aj negatívnu. Z vlastnej skúsenosti môžem potvrdiť, že negatívna je ešte o niečo účinnejšia ako pozitívna. Hlavne ak sa týka vecí, na ktorých nám záleží alebo sú úzko späté s našim životom. Pre tínedžerov to môže byť hrozba odobratia vreckového, pre vysokoškolákov neabsolvovanie skúšky, strata internátu alebo dokonca vylúčenie zo školy a pre zamestnancov samozrejme strata zamestnania. Ako správne ľudí motivovať a aké prostriedky pri tom použiť sú otázky, ktoré si manažéri kladú veľmi často a nájsť na ne správne odpovede býva mnohokrát náročná úloha.

Dôsledky nevhodného zásahu

Keďže aj manažéri sú ľudia, z času na čas môže ich počínanie napáchať viac škody ako úžitku. Stačí ak sa im nepodariť zhromaždiť dostatočné množstvo relevantných informácií alebo sa nachádzajú v časovej tiesni a nesprávne rozhodnutie je na svete. Aké následky môže ich neprimeraný zásah spôsobiť?

Predstavme si učebnicový príklad, kedy sa neskúsený manažér pokúša zachrániť projekt, ktorý už prekročil dohodnutý termín odovzdania, pridaním nových členov do vývojového tímu. Tento zásah neprinesie pre projekt okrem zvýšenia nákladov a ešte neskoršieho odovzdania žiadne zlepšenie. Vysvetlenie je jednoduché. Už v roku 1975 tento jav zdokumentoval Fred Brooks a uviedol nasledujúce dôkazy:

1. ľuďom pridaným do projektu trvá nejaký čas, kým začnú byť produktívnymi. Je potrebné ich zaškoliť a na tento proces sa využijú zdroje, ktoré by za normálnych okolností mohli na projekte pracovať
2. noví odborníci môžu dokonca pôsobiť kontraproduktívne, ak sa im napríklad podarí objaviť nové chyby v existujúcom systéme, ktorých náprava ešte viac oddiali dátum odovzdania
3. pridaním ľudí sa automaticky zvýši počet existujúcich komunikačných kanálov a to dokonca exponenciálne. Čím viac ľudí sa podieľa na plnení určitej úlohy, tým viac času trávia zisťovaním informácií o tom, čo kto robí [22]

Ako ďalšie príklady nie šťastných zásahov do procesu vývoja by som spomenul neprimerané zvýšenie počtu inšpekcií zamestnancov, výmenu vedúceho tímu v neskoršej fáze vývoja alebo neprimerane ostrá kritika jedného alebo viacerých členov tímu.

Záver

Monitorovanie sa stalo nenahraditeľnou súčasťou manažmentu softvérového projektu, bez ktorého sa nezaobíde žiaden kvalitný projekt. Analytikmi navrhnutý plán tvorí základ, bez ktorého by sa monitorovanie nemalo o čo opierať. Ak bolo na vytvorenie plánu vynaložené vysoké úsilie, sily vložené do monitorovania by mali byť o to väčšie, aby

manažéri doviedli projekt do úspešného konca. Neustále sa objavujú pokusy o zefektívnenie procesu monitorovania, ktorý sa počas minulých rokov dostal na úctyhodnú úroveň. Cieľom tejto eseje bolo vyjadriť môj názor k viacerým fázam procesu monitorovania ako takého. Dúfam, že sa mi to aspoň z časti podarilo a táto eseja ponúkne jej čitateľom zopár zaujímavých pohľadov na danú problematiku.

Použitá literatúra

1. BOFFOLI, N., BRUNO, G., CAIVANO, D., MASTELLONI, G.: *Statistical Process Control for Software: A Systematic Approach*. ESEM '08: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. ACM., 2008.
2. BROOKS, F.: *The Mythical Man Month: Essays on Software Engineering*. 2. vyd. Addison-Wesley Profesional. 1995.
3. CANGUSSU, J. W., DECARLO, R. A., MATHUR, A. P.: *Monitoring the Software Test Process Using Statistical Process Control: A Logarithmic Approach*. In: ESEC/FSE-11: Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering. 2003.
4. ZHAO, F., PENG, X., ZHAO, W.: *Software Development Process Monitoring Based on Nominal Transformation*. Eighth IEEE/ACIS International Conference on Computer and Information Science (2009).

Annotation

Monitoring of software project – goals and approaches

Monitoring the state and progress of the project is an important part of software project management. Without monitoring, deploying the software to the customer on time and within budget would be only dream of the software team. Since the project manager gets plan of software project progress, his most important task is to track and monitor work of all teams included in project. The goal of this difficult activity is to identify problem as soon as possible in any phase of the project, and intervene to reduce possible damages. In my paper, I am summarizing limitations, that affect relevancy and accuracy of the planning, which is the background for monitoring the project. Next I am focusing on summarizing goals and methods of monitoring process itself, together with identifying the reasons of aroused anomalies and the way of fixing them. In the end I focus on the possible consequences of incorrect interventions to software project progress.