

Úskalia monitorovania softvérového projektu

RADOSLAV HALAMA

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
Dlizzy[zavináč]centrum[.]sk*

Abstrakt. Každý sebaistý projekt si musel na začiatku stanoviť požiadavky a ciele. Práve proces monitorovania slúži ako mechanizmus na kontrolu týchto cieľov, etáp a zdrojov projektu, ako sú financie, ľudia, čas a kvalita. Tento proces taktiež pomáha projektovým manažérom dozvedieť sa ako projekt napreduje a zároveň dáva skoré varovanie pred možnými problémami a ťažkosťami, ktoré sa počas životného cyklu objavajú. Cieľom eseje je vysvetliť, prečo vlastne monitorovať a následne načrtnúť úskalia samotného procesu monitorovania, ako je napríklad subjektivismus v posudzovaní napredovania jednotlivých etáp v projekte a subjektivismus v posudzovaní práce samotných ľudí. Práca sa taktiež zaoberá reakciami na vzniknutý problém a to najmä z hľadiska pridania nového člena do tímu alebo znížením požiadaviek na vyvíjaný systém.

Úvod

Projekt je odštartovaný, komunikácia prebieha výborne a často, veci nabrali správny smer – takže si môžeme sadnúť a relaxovať. Omyl!

Prácu na projekte možno prirovnať k točeniu tanierov na tyčiac. Je potrebné neustále sledovať každý tanier a v prípade, že by mohol spadnúť, zareagovať či už to jeho spomalením, alebo postupným roztočením na správnu rýchlosť. Monitorovanie v procese vývoja softvérového systému zodpovedá práve sledovaniu tanierov, najmä tých, ktoré majú tendenciu spadnúť. Jeho úlohou je taktiež reagovať na vznikajúce problémy.

Monitorovanie možno teda jednoducho opísať ako proces dozerania na všetky činnosti v projekte a ich prípadnej nápravy.

Monitorovanie postupu projektu, dozeranie na oblasti, v ktorých je najpravdepodobnejší výskyt problému a očakávanie možných ťažkostí predtým ako sa naozaj objavajú, môže ušetriť mnoho času a zdrojov.

Potrebu monitorovania možno ešte viac zvýrazniť nasledujúce štatistiky prieskumov *CHAOS Report* a *Bull Survey* [6], ktoré sa zaoberali neúspešnými projektami a príčinami ich neúspechu.

Zo štatistík Bull Survey vyplýva, že:

- nedostatkom komunikácie trpí až 57% projektov
- nedostatočným alebo žiadnym plánovaním 39% projektov
- zlou kontrolu kvality 35% projektov
- zlým rozpočtovým manažmentom 29% projektov
- a zlým celkovým manažmentom 20% projektov.

Ešte zaujímavejšie sa môžu zdať údaje z CHAOS Report, ktoré hovoria, že:

- 31,1% projektov bolo zastavených pred svojim dokončením
- 52,7% projektov prevýšilo svoje výdavky na 189% pôvodného rozpočtu.

Z uvedených štatistík mi vyplynulo, že minimálne problémy, ako je komunikácia, kontrola kvality a manažment financií, možno monitorovaním čo najskôr zachytiť a následne riešiť a ušetriť tým finančné prostriedky, ktoré sú hlavnou motiváciou každého komerčného softvérového projektu. Taktiež si myslím, že monitorovanie a najmä jeho výsledky môžu pomôcť projektovému manažérovi pri rozhodnutiach v nasledujúcom alebo niektorom z nasledujúcich projektov.

Všeobecný pohľad na úskalia monitorovania

V nasledujúcich kapitolách sa budem zaoberať faktom, že aj keď sa rozhodlo, že sa projekt bude monitorovať, tak to ešte neznamená, že sa mu zaručene podarí dôjsť úspešne do cieľa.

Jedným z hlavných úskalí je fakt, že sa jedná z veľkej časti o subjektívny proces, či už v posudzovaní napredovania projektu, hodnotení ľudí, alebo návrhu riešenia vzniknutého problému. Objektívne rady, ako postupovať pri riešení, doposiaľ neexistujú a osobne si myslím, že ani existovať nebudú, keďže cieľom projektu je vytvoriť vždy niečo nové, doteraz neexistujúce a jedná sa o proces, v ktorom sa vyskytuje veľké množstvo premenných, ako sú ľudia, čas, financie, požiadavky alebo kvalita, ktoré sú odlišné v každom projekte.

Podľa [5] spočíva náročnosť softvérového projektu a teda aj jeho monitorovania vo vlastnostiach samotného softvéru, ako sú:

- *Neviditeľnosť* – fakt, že softvérový produkt je na rozdiel od hardvéru nehmatateľný. Nemožno jednoducho určiť alebo merať jeho kvalitu.
- *Zložitosť* – softvérové produkty sú ťažko pochopiteľné často aj pre samotných vývojárov, čo so sebou prináša komunikačné, technické a manažérske problémy.

- *Nestálosť požiadaviek* – softvér je na rozdiel od hardvéru ľahšie meniteľný, takže aj požiadavky naň sú často predmetom zmeny, s ktorými treba rátať už počas vývoja.

Problém sledovania postupu jednotlivých etáp vývoja

Osobne si myslím, že najlepším spôsobom predchádzania ťažkostiam v projekte, je monitorovať celý životný cyklus vývoja systému, od špecifikovania požiadaviek po nasadenie do prevádzky. Tu sa však vynára problém, ako to monitorovať. Každá fáza životného cyklu je totiž odlišná a má iný cieľ.

Postup a cieľ

Ako už bolo spomenuté, jedným z cieľov monitorovania je sledovať jednotlivé etapy vývoja a ich napredovanie.

Podľa [3] je vyjadrenie napredovania subjektívne, pretože neexistuje univerzálny ideál na to, ako posúdiť, či sa veci zlepšujú alebo nie. [3] ako príklad uvádza príklad s kopaním jamy, kde prvý projekt má za úlohu vykopať jamu a druhý projekt má za úlohu zahrabať inú jamu. Odoberanie zeme môže byť v jednom prípade považované za postup v práci, avšak v druhom prípade za opak.

Meranie postupu v nejakej etape je spojené s cieľom etapy. Postup vyjadruje, najčastejšie v percentách, ako ďaleko sme v dosiahnutí cieľa.

Tu sa objavuje ďalší z problémov. Tým je, že nemusíme mať dobre stanovený cieľ. V takomto prípade môže byť postup plynulý, zodpovedať plánu. Vývojári si môžu myslieť, že projekt napreduje, ale v skutočnosti len stratili čas, lebo výstup etapy nebude pre projekt užitočný, dokonca môže byť až škodlivý.

Pre niektoré etapy vývoja je dokonca len veľmi ťažko povedať, čo je cieľom, alebo presnejšie, kedy je cieľ už splnený. Jedná sa o fázy špecifikovania požiadaviek, analýzy a návrhu.

Fáza špecifikovania požiadaviek

Pre fázu špecifikovania požiadaviek je cieľ teoreticky jednoduchý a to špecifikovať celkovú funkcionálnu kapacitu systému. Častokrát však ani samotný objednávateľ nevie čo chce alebo potrebuje. Svoje požiadavky mení aj v neskorších fázach vývoja.

Fáza analýzy

Cieľom fázy analýzy je oboznámiť sa s problematikou, poprípade s inými vedomosťami, ktoré by mohli byť využité pri tvorbe systému. Ako však merať napredovanie v tomto prípade? Nie je možné určiť, v koľkých percentách progresu tejto fázy sa vývojári už nachádzajú, keďže oblasť, s ktorou sa oboznamujú je pre nich nová a rôzni členovia tímu majú rôzne úlohy. Zaznamenávať napredovanie by bolo možné v prípade, že by existovala akási príručka alebo dokumentácia, ktorá by obsahovala všetky potrebné informácie a po ktorej prečítaní by sa mohlo povedať, že

sme dosiahli cieľ. Postup by sa teda mohol merať na základe prečítaných strán. Tento systém sa mi však zdá nepravdepodobný a mohlo by sa napríklad stať, že záverečné strany by boli náročnejšie na porozumenie. Prečítanie posledných dvadsiatich strán by mohlo byť rovnako časovo náročné, ako prečítanie prvých tristo strán.

Fáza návrhu

Vo fáze návrhu je sledovanie napredovania taktiež náročné. Mohlo by byť merané na základe toho, koľko požiadaviek zákazníka by už na základe implementácie aktuálneho stavu návrhu mohlo byť naplnených. Takáto metrika by však nevyhovovala z dôvodu, že etapa návrhu je špecifický proces, ktorý sa často zaoberá spĺňaním jednotlivých požiadaviek až v neskorších fázach, pričom väčšina času je venovaná architektúre systému a jeho elementov, ktoré sa nemusia priamo podieľať na niektorej z funkcionalít.

Ako som načrtol, je zrejmé, že komplexné monitorovanie prvých fáz vývoja softvéru je náročné a často zradné. Prečo ho teda nezamerať len na fázy implementácie a testovania? Jednou z príčin, prečo sa podľa mňa oplatí aj napriek spomenutým ťažkostiam monitorovať kompletný proces, je napríklad *Aschov efekt*. Ten opisuje jav, keď sa niektorí členovia tímu podriadia nesprávnemu názoru väčšiny, hoci jej úsudok je nesprávny. Kritickou pre tento efekt sa javí najmä fáza plánovania a návrhu.

Fáza implementácie

Asi najväčšia časť monitorovania je zameraná na fázu *implementácie*. Je to fáza, kde systém nadobúda reálny obraz a získava svoju funkcionálnu. Podľa Bercuna [1] je dôležité najmä tu odlišovať prácu a napredovanie. Bercun ďalej uvádza, že prácu možno merať podľa:

- odpracovaných hodín
- napísaných správ
- uskutočnených telefónnych hovorov
- alebo naprogramovaných riadkov kódu.

Týchto metrik sa však treba vyvarovať, lebo žiadna z nich neponúka pohľad na to, ako skutočne projekt napreduje a môžu byť preto zavádzajúce.

Bercun ponúka aj lepší spôsob, pomocou ktorého možno presnejšie identifikovať aktuálny stav projektu a následne ho porovnať s naposledy monitorovaným stavom, alebo so stavom, do ktorého sa chceme dostať a získať tak lepší pohľad na postup projektu. Bercun odporúča projektovému manažérovi klásť členom tímu nasledujúce otázky:

1. O koľko bližšie nás posunula vaša dnešná práca bližšie k dosiahnutiu cieľa?
2. Bola vykonaná práca dostatočne kvalitná?
3. Akú prekážku ste dnes prekonali?

Z môjho pohľadu je však tento prístup veľmi subjektívny, keďže na otázky o svojej produktivite odpovedá samotný zamestnanec. Myslím si, že najvhodnejšie by z tohto pohľadu bolo spojenie medzi oboma prístupmi a teda merať prácu napríklad prostredníctvom riadkov kódu, alebo funkčných blokov a následne bližšie určiť, o koľko sme postúpili, prostredníctvom vyššie uvedených otázok.

Zvolenie vhodnej metriky je výhodné nielen pre manažéra, ale aj pre jednotlivých zamestnancov z pohľadu, že im dáva možnosť sledovať napredovanie celého projektu. Najväčším problémom je však v tomto smere podľa mňa jav, že ľudia prispôbia svoju prácu metrike na úkor cieľa a to tak, aby podľa metriky dosahovali neustály progres, hoci skutočný postup projektu smerom k cieľu je omnoho menší.

Kritické body monitorovania výkonu členov tímu

Monitorovanie výkonu jednotlivých členov tímu je dôležité z viacerých hľadísk:

- zistenie aktuálneho stavu projektu
- plánovanie budúcich aktivít
- interné ohodnotenie zamestnancov
- pomôcka pre odmeňovanie zamestnancov
- pomôcka pre nasadenie do iných projektov.

Jedná sa o mimoriadne citlivý proces, keďže sa týka samotných ľudí, častokrát priateľov. Je preto dôležité vyvarovať sa chýb. Častá chyba, ktorá sa vyskytuje pri posudzovaní výkonov je, že sa meranie zameriava na kvantitu a nie kvalitu. V takom prípade sa posudzuje úsilie a nie výkon. Za výkon možno označiť tú časť úsilia, ktorá sa efektívne prejaví do postupu v projekte.

Výber metód, na základe ktorých sa bude monitorovať, by mal zodpovedať danej situácii v tíme a je taktiež vhodné, aby ich manažér vopred prediskutoval s členmi tímu. Musí zistiť, či im daný spôsob bude vyhovovať a nebude ich zbytočne zneisťovať a odvádzať od práce.

Dôležitým faktom pre výber metriky je odbúrať *subjektivitu*, ktorá je spojená s každým takýmto hodnotením a môže sa prejaviť napríklad ako preferovanie niektorých jedincov, prehliadanie ich nedostatkov. Zvolenie vhodnej metriky je dôležité aj z hľadiska, aby sa nestalo, že manažér potrestá niekoho za to, že uviedol pravdivé údaje o stave.

Taktiež si myslím, že niektorí jedinci môžu považovať monitorovanie ich výkonu za akési cieleňé naznačenie, že ich práca nie je adekvátne a že manažér s nimi nie je spokojný. Z celkového hľadiska však považujem meranie produktivity pracovníkov za prospešné, pretože pokiaľ nemerame – nemôžeme zlepšovať.

Problematika výberu vhodnej reakcie na vzniknutý problém

Ako bolo spomenuté, proces monitorovania zahŕňa okrem objavovania chýb projektu aj reakciu na ne a teda riešenie vznikajúcich problémov. Táto časť je podľa mňa prinajmenšom rovnako dôležitá, ako fáza kontroly a dopustiť sa omylu v tejto fáze by znamenalo zbytočnosť celej dovtedajšej monitorovacej činnosti.

Ak sa na problém výberu riešenia pozeráme z makro hľadiska, tak možností nie je až tak veľa, ako by sa mohlo zdať. Jenkins [4] vo svojej práci opisuje, že projekt má tri základné entity, ktoré spolu so svojimi vzťahmi možno zobraziť do takzvaného *trojuholníka kvality* (Obr. 1), v ktorom:

- čas (time) predstavuje čas, ktorý je dostupný do odovzdania projektu.
- cena (cost) predstavuje všetky dostupné financie a zdroje projektu.
- kvalita (quality) reprezentuje množstvo požiadaviek, ktoré musí projekt naplniť, aby bol úspešný.



Obr.1. Trojuholník kvality – Scope triangle [4].

Bežnou situáciou je jav, keď jeden z faktorov je konštantný. Z trojuholníka potom vyplýva, že ostatné dva faktory sú na seba priamo úmerné. Ak je napríklad presne určená doba odovzdania, kvalita ide ruku v ruke s cenou. Čím viac financií je pre projekt uvoľnených, či už na zaplatenie väčšieho počtu pracovníkov alebo iných zdrojov, tým kvalitnejší by mal byť projekt.

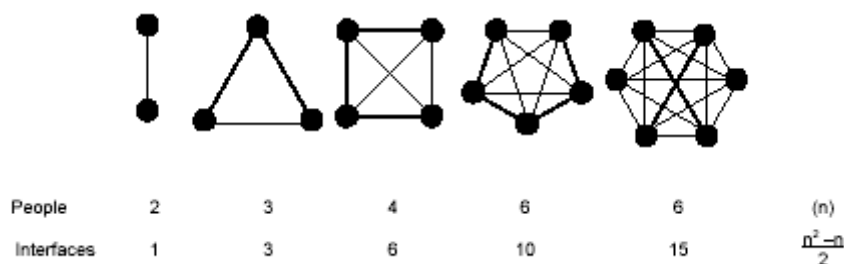
Ak by sa v takomto projekte vyskytli chyby, ktorých odstránenie by za daných okolností predĺžilo čas odovzdania, existujú len dve možné riešenia a to *pridanie ďalších zdrojov* v podobe pracovníkov, alebo *zníženie kvality*, čo sa prejaví znížením funkcionality vyvíjaného systému.

Možno teda povedať, že projektový manažér má všeobecne pri indikácii problému na výber z troch možností:

- pridať čas – odovzdať projekt neskôr a získať tak čas na dokončenie funkcionality.

- zvýšiť náklady – pribrať do tímu ďalších pracovníkov na rýchlejšie ukončenie projektu.
- znížiť kvalitu – implementovať len hlavnú funkcionality.

Podľa Brooksa [2] je však práve možnosť pridania ďalšieho člena do tímu tou najzradnejšou možnosťou. Brooks argumentuje, že hoci je pravda, že viac zdrojov urýchli vývoj, zrýchlenie nie je priamo úmerné počtu pridaných ľudí. Hlavným problémom je nárast zložitosti interakcie a komunikácie, narastajúcim exponenciálne s každým novým členom tímu (Obr. 2). Vrcholy grafu môžu predstavovať jednotlivcov, skupinu ľudí, ale aj celé tímy alebo organizácie.



Obr.2. Exponenciálny nárast interakcií s každým pridaným človekom [4].

Ďalším faktorom toho, že práca na projekte sa neurýchli je aj fakt, že každá nová osoba bude musieť byť zaučená a oboznámená s cieľmi projektu, pričom zaučiť ju bude musieť jeden z členov tímu, ktorý tým stratí svoj čas, ktorý by mohol venovať priamo projektu. Taktiež mi pripadá zrejmé, že nová osoba nebude produktívna ihneď po nasadení do tímu, čo mi príde ako ďalšie mínus pre krátkodobé projekty.

Dostávam sa teda opäť k otázke výberu vhodnej reakcie na vznikajúci problém. Ak o možnosti pridania nových členov do tímu budem na základe vyššie uvedených faktov uvažovať ako o okrajovom riešení, najvhodnejšou možnosťou sa mi javí požiadať o predĺženie doby dodania systému. V prípade, že je čas dodania fixne určený zmluvou, musí dôjsť k zníženiu funkcionality. Osobne si myslím, že sa jedná o oveľa menší problém ako nedodať žiadny systém. Dodaný softvér by splňal základné požiadavky a dodatočná funkcionality by sa mohla k systému pridať v jednom z následných *update*-ov systému.

Záver

Esej sa snaží poukázať na potrebu, ale najmä náročnosť procesu monitorovania, kde pre veľké množstvo faktorov neexistuje optimálny model, ktorým možno merať stav softvérového projektu vyvíjaného v tíme. Aj tie najmodernejšie metódy sa aspoň z časti opierajú len o subjektívne odhady členov tímu. Úspech je teda vo veľkej miere závislý na schopnostiach a predošliých skúsenostiach manažéra a členov tímu.

Z celkového hľadiska však považujem monitorovanie za prospešné. V prvých projektoch možno prinesie viac problémov ako prospechu, no z dlhodobejšieho hľadiska je pre firmu výhodné zabudovať tento mechanizmus do svojich projektov a to najlepšie čo najskôr, pretože pokiaľ nemeriam, nemôžem zlepšovať.

Použitá literatúra

1. Bercun, S.: *Work vs. Progress*, [online]. August 2005. Dostupné na: <<http://www.scottberkun.com/essays/45-work-vs-progress/>>
2. Brooks, F. P.: *The Mythical Man-Month (Anniversary Ed.)*. Addison-Wesley Longman Publishing Co., Inc (1995), 13-26.
3. IFPUG: *How to effectively track software progress*. [online]. 28. januára 2002. Dostupné na: <<http://www.informit.com/articles/article.aspx?p=27356>>
4. Jenkins, N.: *A Project Management Primer or a guide to making projects work*. [online]. 2006. Dostupné na: <<http://www.exinfm.com/training/pdfiles/projectPrimer.pdf>>
5. Jurison, J.: *Software project management: the manager's view*. *Commun. AIS* 2, 3es (Nov. 1999), 2.
6. Wood M.: *Project Failure Statistics and Facts*. [online]. September 2008. Dostupné na: <<http://www.pmhut.com/project-failure-statistics-and-facts>>

Annotation

Troubles of software project monitoring

Each potential project has to set up its requirements and goals. Recently, the process of monitoring serves as a control mechanism for all the goals, activities and project sources, such as finances, people, time and quality. This process also helps project managers to find out how the project is progressing and also gives early warning of all possible problems and difficulties, which occur during the life cycle of software development. The aim of essay is to explain why to monitor and then outline the challenges of the process of monitoring, such as subjectivity of tracking progress in various stages of project and subjectivism in assessing the work of people. Essay also deals with responses to problems, especially from view of adding the new member to the team, or cutting exceptional functionality of system.