

# MONITOROVAŤ ? ÁNO, URČITE ÁNO !

*Kto sa nič nespýta, nič nevie.*

*Anton Kráľovič*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
Kralovic.anton[zavináč]gmail[.]com

**Abstrakt.** *Monitorovanie ako také v praxi znamená pravidelné sledovanie akéhokoľvek progresu v softvérom projekte. Je to činnosť ktorá má primárne dopomáhať manažérovi softvérového projektu k včasnému odhaleniu akýchkoľvek nedostatkov na jeho projekte, či už nesplnenie vopred vytvoreného časového plánu alebo nesplnenie zákazníkovoých požiadaviek. Takéto monitorovanie má samozrejme svoje výhody a aj nevýhody. Aké to sú a aký je ich pomer? Ako sa softvérové projekty vôbec delia a ako sa dá merať ich „spoľahlivosť“? V tejto eseji sa pokúšam odpovedať aj na tieto otázky.*

**Kľúčové slová:** *softvérové metriky, metódy monitorovania, SMF, outsourcing, opensource*

## Úvod

Cieľom každého dobrého projektového manažéra je určite to aby bol jeho projekt úspešný. K tomu potrebuje projektový manažér byť schopný efektívne riadiť svoj projekt a rýchlo ale hlavne korektne reagovať na každý vzniknutý problém. Pretože existuje mnoho nekontrolovateľných vplyvov, ktoré môžu ovplyvniť vývojový proces. Napríklad zákazník môže požadovať dodatočnú zmenu výstupných požiadaviek, v ktorejkoľvek fáze projektu môžeme naraziť na závažný problém ktorý vyžaduje viac času na vyriešenie a podobne. Práve takýmto chybám sa snažíme predísť monitorovaním projektu.

Dôležitosť monitorovania projektu je podložená štatistikami a to že aj napriek najnovším metodikám vo vývoji softvéru skoro 80 % projektov nedodrží korektný čas ukončenia alebo v horšom prípade na korektné ukončenie projektu sú potrebné ďaleko väčšie náklady ako bolo uvažované v pôvodnom rozpočte [1].

## Prečo a vôbec ako monitorovať ?

Projektový manažér by mal byť zodpovedný za to aby boli všetky prebiehajúce procesy vykonávané na čas, to znamená podľa vo vopred vypracovaného časového plánu a zároveň aby pri vykonávaní týchto procesov projekt nezačínal prevyšovať rozpočet. Počas životného cyklu každého projektu je manažér projektu konfrontovaný s mnohými problémami, či už technickými, personálnymi, alebo takými ktoré sa týkajú priamo niektorého manažmentu. Všetko toto je spôsobené faktom že vývoj softvéru je proces dynamický a preto naň vplývajú rôzne faktory ako neplánované zmeny požiadaviek na systém, na personál alebo na technológiu. Všetky merania ktoré sa na projekte budú vykonávať musia byť preto taktiež vykonávané dynamicky, aby mal vedúci celého projektu alebo manažér monitorovania k dispozícii všetky využiteľné informácie z týchto meraní v reálnom čase.

Na monitorovanie procesov v projekte je potrebné aby sa pri každom projekte používal vopred pripravený systém, boli vyčlenené osoby zodpovedné za monitoring a v neposlednom rade aby sa v čo najväčšej miere využívali podporné nástroje. Tieto nepísané pravidla by mali podľa mňa dodržiavať už stredné a samozrejme veľké softvérové projekty alebo firmy. Miera monitoringu je určite silno ovplyvnená veľkosťou ale hlavne typom projektu, ktorým sa budem venovať nižšie. V projektoch menšieho alebo menej náročného charakteru nie je väčšinou potrebné zavádzanie komplikovaných monitorovacích metód a v tomto prípade by to mohlo skôr vytvoriť sekundárny efekt tzv. „waste of time“ (angl. strata času). Tu je možné použitie jednoduchého podporného nástroja na synchronizáciu jednotlivých modulov vyvíjaného softvéru.

### Typy projektu a ich riziká

Počas doterajšieho vývoja v oblasti softvérového inžinierstva vzniklo niekoľko prístupov monitorovania softvérového projektu, ktoré sú viac či menej vhodné pre daný typ projektu.

Väčšina z nich popisuje postup pri monitorovaní *klasických typov projektov* kde je základom úzka komunikácia v tíme čo zabezpečuje dobrý prehľad projektového manažéra o aktuálnom stave projektu. Avšak podľa toho čo tvrdím v predošlej vete musíme logicky naraziť na jeden veľký problém. Asi najväčšie riziko predstavuje fakt že zainteresovaný ľudia a tí často kráť nemusia z rôznych dôvodov podávať nesprávne alebo neobjektívne informácie a zamlčovať vyskytnuté problémy. Aby sa takýmto rizikám dalo predísť boli vyvinuté rôzne metriky na meranie progresu v softvérových projektoch, ktorým sa tiež ešte budem v eseji venovať.

Naopak pri *outsourcovaných projektoch* sa nemusia všetci zainteresovaný do projektu nachádzať na jednom pracovisku, taktiež môžu byť jednotlivé moduly softvéru vytvárané rôznymi tímami dokonca aj na rôznych kontinentoch. Tým pádom projektový manažér nemá priamy dosah na všetkých účastníkov projektu. Z toho je zrejme že pri tomto type projektov implicitne rastie náročnosť spravovania projektu s pribúdajúcim počtom tímov a ľudí. Ako keby to nestačilo tak jednotlivé úlohy pre jednu stranu alebo jeden tím môžu priamo ovplyvňovať úlohu alebo úlohy pre iný tím. Preto je v tomto prípade veľmi dôležité monitorovanie stavu úloh, pričom treba dbať na synchronizáciu medzi jednotlivými tímami.

Špecifickým typom projektu je *open-source* projekt. Pri tomto type projektu mnoho krát pracujú ľudia úplne nezávisle od seba a vôbec nikto ich nesleduje. Preto je potrebné v takomto projekte sledovať jeho priebeh aby sme boli schopný čím skôr odhaliť prípadné ohrozenie projektu. Pri *open-source* projektoch je väčšina úloh zvyčajne vykonávaná neplatenými účastníkmi. Z toho je podľa mňa jasné ako to niekedy ovplyvňuje kvalitu a dokonca v mnohých prípadoch spôsobuje častú obmenu účastníkov. Tieto projekty sú taktiež charakteristické svojou nižšou mierou formálnosti alebo dizajnu a omnoho menším úsilím vynaloženým na plánovanie a manažment. Ako aj v predchádzajúcich dvoch prípadoch tu najväčšie riziko predstavujú zúčastnení ľudia, ktorý musia v pravidelných časových intervaloch podávať aspoň nejaký report a hlásiť vzniknuté problémy. Na pomoc manažérovi existujú ale niektoré indikátory, podľa ktorých môžeme posúdiť progres alebo aktuálny stav v *open-source* projekte a to napríklad [4]:

1. Intenzita aktivity komunity – interakcia medzi účastníkmi, vývojármi
2. Proporcie – súčet komentárov a diskusií v mailing liste, zmeny v SVN systéme
3. Open problems and delayed services (angl. otvorené problémy a meškajúce služby) – registrované problémy v systéme pre ich správu ktoré neboli vyriešené do stanoveného termínu

### Výber metódy monitorovania

Informácie, ktoré sú zbierané počas životného cyklu projektu, nech už sú zbierané akokoľvek, v konečnom dôsledku potrebujú byť nejako využité. Samozrejme analogicky by mala nasledovať analýza. Avšak prečo by sme sa mali trápiť, keď môžeme využiť už vytvorený model, ktorý identifikuje hlavné kontrolné body a použiť ich pri analyzovaní našich informácií. Uvažujme o klasickom trojrozmernom modeli softvéru, v ktorom jednotlivé rozmery definujú čas, cenu a kvalitu produktu. Tieto aspekty sú úzko spojené s aktivitou účastníkov na projekte, dodržiavaním projektového plánu, funkcionalitou softvérového produktu, zapojenými ľuďmi a ich nákladmi a podobne [3].

Práve podľa týchto prvkov dostal pomenovanie *AHA* (angl. Activity Human Artifact). Použitie kontrolných bodov je veľmi prospešné v každej fáze projektu. Kontrolné body môžu napríklad riešiť aktuálne dodržiavanie časového plánu projektu, dozerať na aktívnu komunikáciu v tíme, taktiež môže riešiť vplyv iných udalostí na hlavné body ako napríklad použitie nesprávnej technológie bude mať za dôsledok oneskorenie v pláne a prekročenie nákladov.

Práve na zvládanie podobných problémov sa zameriava *SMF* (angl. Software Monitoring Framework) spomenutý v literatúre [2]. Tento rámec na monitorovanie softvéru nám ponúka komplexnejší spôsob, ako pristupovať k monitoringu. Aktivity, ktoré sú vykonávané počas vývoja softvérového projektu rozdeľuje do dvoch základných fáz, *akvizíčnej a operačnej*.

Hlavnými piliermi *akvizíčnej fázy* sú odhad potrebného úsilia, vytvorenie systému na jeho manažment, vytvorenie databázy údajov o predošlom úsilí a odhalenie kritických faktorov vývoja. Je všeobecne známe že správny odhad je veľmi citlivá záležitosť pretože v praxi si uvedomujeme chybu väčšinou neskoro. Aj keď existujú rôzne matematické metódy, veľmi dôležitý je subjektívny odhad. To robí manažment plánovania náročným hlavne pre menej skúsených manažérov. Aby bol odhad správny, musia čo najlepšie

vedieť identifikovať všetky procesy, ktoré budú základom pre správny životný cyklus projektu a tiež rozpoznať procesy ktoré pohltia najviac zdrojov, či už časových alebo finančných. Okrem takéhoto odhadu by podľa mňa manažér mal určiť miesta, kde by sa mohol projekt prípadne modifikovať. Je potrebné tiež odhadnúť pravdepodobnosť výskytu takýchto miest a riešiť ich dopad na projekt ako taký.

Pre účel monitorovania je nutné stanoviť v rannej fáze projektu všeobecné pravidlá. Tieto sa okrem iného budú zaoberať správnym výberom metrík a štandardov na meranie výdavkov a podaných výkonov účastníkov. V jednotlivých diskusiách medzi členmi tímu je si myslím potrebné udržiavať istú všeobecnosť a neriešiť hlbšie také problémy z celým tímom pokiaľ nie sú do konkrétneho problému zainteresovaný všetci.

Na monitorovanie celého procesu vývoja softvéru ponúka SMF možnosť sledovania pokroku v projekte pomocou míľnikov. V jednotlivých fázach vývoja sa stanovujú takéto míľniky a v nich sa zhodnocuje vykonaný alebo aj nevykonaný progres, napríklad po implementačnej alebo testovacej fáze.

Akvizičná fáza slúži teda na uskutočnenie všetkých spomenutých odhadov a nasledujúce monitorovanie vývoja projektu. Myslím že je dôležité pripomenúť že všetky vyššie spomínané aktivity je treba vykonať maximálne dôsledne, pretože čo i len malé zaváhania alebo odchýlka môžu mať v konečnom dôsledku veľmi nepríjemný dopad na projekt.

V *operačnej fáze* sa už používajú rôzne metriky na sledovanie pokroku v aktivitách projektu. Autori SMF zvlášť dávajú do pozornosti neformálne monitorovacie mechanizmy, ktorými sú stretnutia s rôznymi tímami počas priebehu vývoja projektu a intenzívna komunikácia, ktorá veľmi často pozitívne prispieva k odhaleniu vznikajúcich problémov. Hlavnou úlohou operačnej fázy je podať projektovému manažérovi informácie o tom, aké problémy v skutočnosti nastali. Manažér sa pozrie na plánovaný rozvrh, porovná ho so skutočnými časmi trvania procesov a sám môže usúdiť či stále vie dodržať stanovený čas odovzdania alebo bude musieť prijať nejaké opravné opatrenia.

Použitie SMF neodstráni výskyt problémov ale má za úlohu odhaliť ich ešte v ich rannom štádiu vzniku a tak dať manažérovi možnosť promptne reagovať a prijať opravné opatrenia čo najskôr, tak aby vážnejšie nepoškodili celý projekt.

## Záver

V tejto eseji som sa snažil porovnať podľa mňa hlavné typy softvérových projektov a rôzne možnosti prístupu k ich monitorovaniu. Dospel som k záverom že čo sa týka klasických projektov tak úspech alebo neúspech môže určovať vhodne zvolené meranie odvedenej práce, použitie vhodnej metriky a dostatočné rozdelenie úloh na podúlohy za účelom lepšieho a ľahšieho monitorovania.

Ostatné dva typy majú podľa môjho názoru podobné prístupy, kde je potrebné kľásť dôraz na sledovanie jednotlivých úloh medzi tímami, prípadne sledovanie udalostí, ktoré chápeme ako interakcie medzi účastníkmi daného softvérového projektu. Pre získavanie relevantných informácií je potrebné vopred definovať nejaký monitorovací systém a správne definovať relevantné udalosti.

Ďalej boli analyzované metódy prístupu k monitorovaniu projektov. Obidve sa snažia o celkové riešenie prístupu k monitorovaniu. SMF však podľa môjho názoru k tejto

problematike pristupuje komplexne, ponúka viacero možností a ich variácie na úspešné monitorovanie projektu. Z môjho pohľadu však dostatočne neurčuje pokrok vykonaný na projekte. Zaujímavou oblasťou na uvažovanie by mohli byť aj alternatívne metódy monitorovania založené na iných ako analytických prístupoch k rozhodovaniu.

## Použitá literatúra

1. Gordon, R. J., Gordon, R. S.: *Information Systems A Management Approach*, The Dryden Press, 1999.
2. Ho-Leung Tsai, Cheung, D.: A monitoring framework for software project development. *Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on* , vol.2, no., pp.1079-1085 vol.2 (1999). 22.10. 2010
3. Juan Li; Nan Jiang; Mingshu Li; Qing Wang; Yanwu Yang: Tracking Projects through A Three-Dimensional Software Development Model, *31st Annual International Computer Software and Applications Conference, 2007*, vol.1, no., pp.301-308 <http://doi.ieeecomputersociety.org/10.1109/COMPSAC.2007.226>
4. Wahyudin, D., Tjoa, A. M.: Event-Based Monitoring of Open Source Software Projects. *The Second International Conference on Availability, Reliability and Security (2007)*, 1108-1115.

## Annotation

*Monitoring? Yes, definitely yes!*

*Monitoring as well as in practice means regular monitoring of progress in any software project. It's activity that has primarily helped by the software project manager for early detection of any shortcomings on his project, whether failure to meet pre-established schedule or not the client's requirements. Such monitoring is of course its advantages and also disadvantages. What are they and what is their proportion? As software projects ever shared and how to measure their "reliability"? In this essay I try to answer these questions.*