

# MONITOROVANIE SOFTVÉROVÉHO PROJEKTU JE AKO STARAŤ SA O MALÉ DIEŤA

*Pri vývoji softvérového projektu nespustite z neho oči*

*Filip Baďura*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
filip.badura@gmail.sk

**Abstrakt.** Každým dňom pracujú tisícky ľudí na vývoji tisíckach softvérových systémov. Aby boli na tieto projekty vynaložené primeraná náklady, aby sa ich vývojom strávilo adekvátne množstvo času, je potrebné tento vývoj projektov kontrolovať. A práve časť manažmentu projektu – monitorovanie nám slúži na to aby sme časom a nákladmi neplytvali. Práve monitorovanie nám pomáha viesť projekt podľa plánu a eliminovať zbytočné problémy pri jeho vývoji. Avšak aj pri monitorovaní si musíme určiť ako správne ho vykonávať, na čo sa zamerať a ktoré metriky použiť.

**Kľúčové slová:** monitorovanie, softvérový projekt, metriky, LOC, funkčné body

## Úvod

Pred začatím samotnej tvorby softvérového produktu býva vo väčšine prípadov vytvorený akýsi plán projektu. Tento plán by mal zahŕňať všetky dôležité vlastnosti projektu, časový harmonogram, rozbor nákladov a podobne. Avšak vytvorenie plánu je len prvým krokom, oveľa dôležitejším a náročnejším je v ďalšej fáze jeho dodržiavanie.

A práve monitorovanie softvérového projektu slúži na to, aby sme vedeli ako plán dodržiavame, koľko sa od neho odlišujeme, a či postupujeme správnym smerom. Tak ako rodičia dohliadajú na svoje malé dieťa a musia ho mať stále na očiach, tak sa treba starať a dohliadať aj na vývoj softvérového projektu. Výchova malého dieťaťa nie je jednoduchá a rovnako tak ani monitorovanie softvérového projektu nie je triviálny problém.

Kým o dieťa sa starajú zväčša dvaja ľudia - rodičia, na vývoji softvérového produktu sa poväčšine podieľa oveľa väčšie množstvo ľudí, čo samozrejme zvyšuje náročnosť pri jeho monitorovaní.

Čo je vlastne monitorovanie projektu ? Pri monitorovaní projektu ide o sledovanie a kontrolu priebehu projektu za účelom uistiť sa, či je priebeh v súlade s rozsahom projektu, harmonogramom a plánmi. Monitorovanie projektu je meranie progresu porovnaním súčasného stavu s plánom projektu a prijímanie potrebných korektívnych opatrení [1].

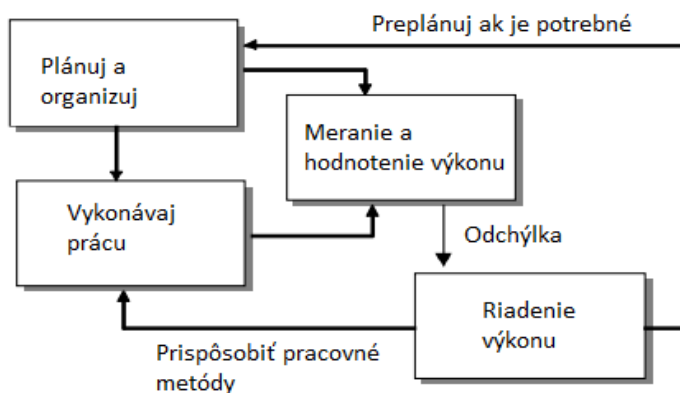
## Prečo monitorovať projekt ?

Nechali by ste svoje dieťa bez dozoru ? Asi nie. Rovnako tak ani vývoj softvérového projektu nemožno nechať bez dozoru a treba ho pravidelne monitorovať. Bez pravidelného monitorovania projektu sa môže veľmi ľahko stať, že nezistená chyba v skorších fázach tvorby softvérového projektu bude mať dopad aj na ďalšie fázy projektu a pre jej odstránenie by sme sa museli vrátiť o niekoľko krokov dozadu. To vo väčšine prípadov spôsobí časový sklz, častokrát zvýšenie nákladov a od toho odvíjajúcu sa nespokojnosť zákazníka.

Projekt monitorujeme jednoducho preto, aby sme sa uistili, či veci idú podľa plánu a v prípade, že tomu tak nie je, aby sme rozpoznali a vhodne reagovali na odchýlky a zmeny oproti plánu.

Pre samotným začatím monitorovania projektu je potrebné uvedomiť si, prečo je táto časť manažmentu softvérového projektu dôležitá a aké sú jej hlavné ciele [2].

- udržiavať a riadiť projekt tak, aby sa v každom momente čo najviac približoval plánu
- identifikovať čo najviac problémov ešte predtým ako nastanú
- nasadiť korektívne opatrenia



Obr.1: Monitorovanie projektu

Tak ako u dieťaťa kontrolujeme jeho výšku, váhu, zdravotný stav a iné dôležité veci, rovnako tak si treba určiť, čo budeme sledovať a monitorovať pri tvorbe softvérového

projektu. Musí ísť o premenné, ktoré si okrem ľudí pracujúcich na projekte dokáže skontrolovať aj samotný zákazník. Preto základnými premennými pri sledovaní projektu sú:

- čas
- náklady
- funkcionálnosť

## Metriky pre monitorovanie softvérového projektu

Nie je ľahké zmerať výšku, váhu, teplotu človeka. Ako však zvoliť správne metriky pre softvérový projekt, aby sme naozaj zistili, či sa ubera správny smerom, a či sa dosahuje požadovaný progres. Bolo by veľmi jednoduché, keby si každý, kto pracuje na projekte určil koľko hodín na ňom pracoval. Otázkou však je, či táto práca naozaj prispela k samotnému progresu projektu. Veľkou súčasťou práce na projekte sú analýzy, komunikácia medzi členmi tímu a podobne. Tieto druhy práce sú pre tvorbu projektu taktiež veľmi dôležitejšie, častokrát možno dôležitejšie než samotná implementácia, avšak nie sú viditeľné a nijak nezasahujú do samotného progresu vývoja projektu.

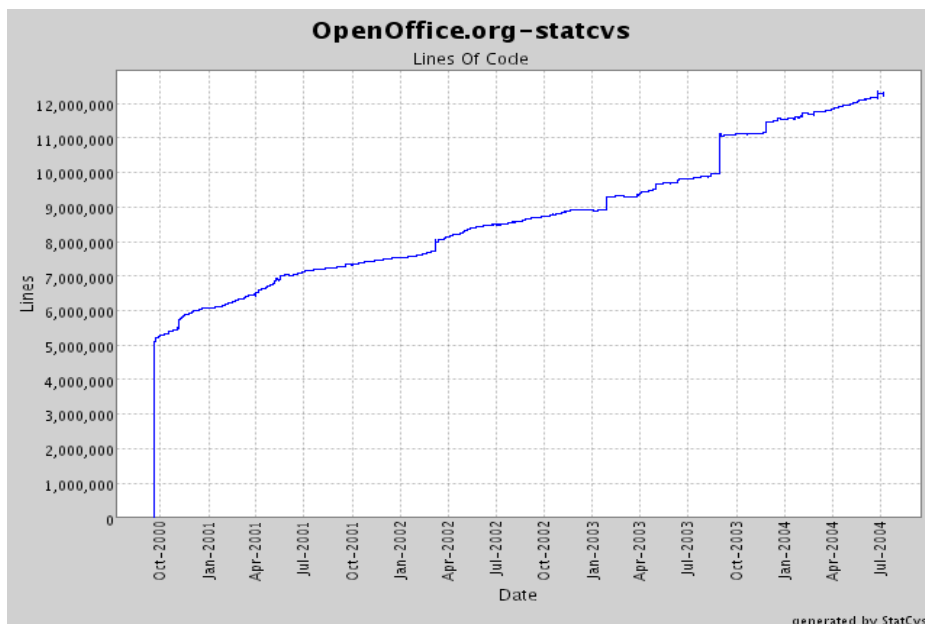
Preto existuje niekoľko metrík, podľa ktorých dokážeme lepšie určiť pokrok vo vývoji softvérového produktu. Každá z týchto metrík má svoje výhody aj nevýhody a ich aplikovanie na projekt nie je vždy rovnako náročné. Taktiež ich výpovedná hodnota nemusí vždy odzrkadľovať presný progres práce na projekte.

Jedna z týchto metrík je meranie projektu pomocou počítania riadkov kódu (LOC – lines of code). Ide o najbežnejšiu metriku s najjednoduchším použitím a možnosťou automatizácie. Avšak pri počítaní riadkov kódu narazíme aj k značným nedostatkom a nevýhodám použitia tejto metriky.

U dieťaťa nie je dôležité iba množstvo jedla, ktoré prijme ale hlavne to aké dôležité látky a vitamíny to jedlo obsahuje. Rovnako tak pri meraní touto metrikou zistíme, že počet nových riadkov kódu nie je vždy priamo úmerný dosiahnutému progresu. Ďalšou nevýhodou podľa zdroja [4] je, že my chceme zistiť koľko vedomostí systém obsahuje alebo bude obsahovať, avšak pri tejto metrike nie je spôsob ako to zmerať.

A tak ako prejedanie sa nie je zdravé, tak poväčšine aj veľké množstvo riadkov nezaručuje automaticky správne smerovanie projektu a dosiahnutie úspechu. Veľké percento riadkov kódu nemá vplyv na funkcionálnosť systému a to znamená, že sú pre zákazníka nepodstatné. Pre neho je dôležitá funkcionálnosť a splnenie jeho požiadaviek a nezaujíma ho koľko času strávil programátor písaním riadkov kódu pre GUI alebo komentárov v programe a koľko tých riadkov vlastne bolo.

Z toho vyplýva, že pomocou LOC sa dá síce merať robusnosť systému a pre samotných programátorov môže táto metrika pomáhať pri monitorovaní projektu ale na druhú stranu len málokedy od počtu riadkov kódu závisí aj jeho funkcionálnosť. To znamená, že v období keď bolo napísaných najviac riadkov kódu nemuselo byť automaticky splnených najviac požiadaviek zákazníka, a taktiež to neznamena, že tieto riadky odrážajú strávený čas pri riešení projektu.



**Obr.2:** Graf nárastu počtu riadkov pri vývoji OpenOffice

Ďalšou metrikou je meranie vynaloženého úsilia analýzou funkčných bodov (function points). Túto metódu vyvinul Allan Albrecht v roku 1979. Ide o objektívny a systematický postup na meranie veľkosti softvéru „vyčíslením,, jeho funkcionality poskytnutej zákazníčkovi, na základe jeho požiadaviek a logického návrhu [4].

Keď je dieťa hladné tak si od rodičov jedlo vypýta. A tak isto keď zákazník požaduje nejakú funkcionality, definuje ju vo svojich požiadavkách. Dieťa je spokojné, keď sa nasýti a zákazník je spokojný keď program pracuje tak ako si on predstavoval a spĺňa vlastnosti, ktoré požadoval.

Tu vidíme hlavnú výhodu použitia tejto metriky. Tou je, že je úzko spätá so zákazníkom a on sám si ju môže kontrolovať. A keďže tím, ktorí má pracovať na softvérovom projekte by mal mať už nejaké skúsenosti vie hneď na začiatku projektu ľahšie odhadnúť potrebný čas a náklady pre splnenie požadovanej funkcionality. Ďalšou výhodou je možnosť vychádzať z predchádzajúcich podobných projektov a podľa nich vykonať odhady a referovať ich zákazníčkovi. To výrazne uľahčuje komunikáciu medzi zákazníčkovi a vývojárskym tímom. Avšak aj táto metrika má svoje nevýhody. Jednou z nich je, že použitie tejto metriky je manuálnou záležitosťou a nie je možné ju zautomatizovať. Druhou nevýhodou je, že pri monitorovaní softvérového projektu a vykonávaní odhadov pomocou funkčných bodov sú potrebné vyššie skúsenosti.

Takže vidíme, že obe metriky majú svoje výhody aj nevýhody a obe môžu byť pri monitorovaní projektu nápomocné. Pre menej skúsený tím je použitie analýzy funkčných bodov zložitý problém a pokiaľ nemajú za sebou podobné projekty tak častokrát aj neriešiteľný. Preto pri vývoji softvérového systému je pre nich výhodnejšie použiť monitorovanie pomocou počítania riadkov kódu. Tu sa však musia dohodnúť na pravidlách pri jeho počítaní (ne/počítame komentáre v kóde, prázdne riadky a pod.),

používať rovnaký programovací jazyk a uvedomiť si jeho nevýhody. Pri vývoji softvérového systému skúsenejším tímom môže analýza funkčných bodov výrazne uľahčiť komunikáciu so zákazníkom, a tím zabrániť zbytočným konfliktom a nejasnostiam.

## Záver

O tom, že monitorovanie projektu počas jeho vývoja je nesmierne dôležité nie je pochýb. Pomáha nám vyhnúť sa problémom v komunikácii so zákazníkom, odhaliť chyby skôr ako by bolo neskoro a taktiež určuje ako sme na tom v porovnaní s plánom. To ako správne projekt monitorovať závisí od viacerých faktorov ako sú veľkosť tímu, jeho miera skúsenosti a nemenej aj od požiadaviek zákazníka.

Dôležité je aby tak ako u dieťaťa boli určité návyky automatické tak aj vykonávanie monitorovania projektu musí byť samozrejmosťou pri jeho vývoji. Toto monitorovanie musí byť pravidelné, tak ako musia byť pravidelné prehliadky človeka u lekára.

„Tak čo, vedeli by ste sa postarať o dieťa?“

## Použitá literatúra

1. <http://www.taskmanagementguid.com/solutions/articles/project-managing-and-monitoring-software.php>
2. Scott Berkun: Work vs. Progress, [www.scottberkun.com/essays/456-work-vs-progress/](http://www.scottberkun.com/essays/456-work-vs-progress/)
3. Kurmanadham V.V.G.B. Gollapudi, Global Microsoft Business Unit, Function Points or Lines of Code? - An Insight.
4. Albrecht, A.J. ;Gaffney, J.E., Jr., Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, Software Engineering

## Annotation

*Every day thousands of people work on development of thousands of software projects. To insure, that a reasonable cost and adequate time are spent on development of these projects, it is necessary to control this development. And it is part of management of project – monitoring, that serves to us, so that we don't waste a time and costs. Monitoring of project helps us to keep project on schedule and avoid unnecessary problems in its development. However, if we want to monitor project, we need to determinate how to do it, what to look for and which metrics to use.*