

# AKÚ METRIKU SI VYBRAŤ?

*Dôveruj ale preveruj!*

Vladimír Sudor

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

**Abstrakt.** Táto práca sa snaží objasniť rôzne metriky monitorovania projektu. Taktiež ponúka dôležité vlastnosti metrík na ktoré treba prihliadať pri výbere konkrétnej metriky. Snažil som sa identifikovať všeobecné vlastnosti, aby ich bolo možné aplikovať na všetky metriky. Ďalej sú v práci spomenuté už konkrétne metriky opísané na základe týchto vlastností. Pri jednotlivých metriky sú stanovené jej výhody a nevýhody. Cieľom práce je určiť čo nám jednotlivé metriky umožňujú a pomôcť čitateľovi lepšie sa rozhodnúť pri výbere konkrétnych metrík. Pretože je metrík veľké množstvo a všetky nemôžeme použiť. Kombináciou vhodných metrík môžeme zamedziť vzniku väčšiny problémov pri vývoji softvéru.

**Kľúčové slová:** metrika, kvalita, efektívnosť, správnosť.

## Úvod

V súčasnej dobe informačné technológie prenikli skoro do všetkých odvetví priemyslu. Vývoj softvéru a samotný softvér je čoraz rozsiahlejší a zložitejší. Projekty softvérových aplikácií sú často realizované veľkými tímami alebo skupinami tímov. To znamená, že vývoj softvéru si vyžaduje stále vyššie náklady.

Preto je nutné proces vývoja softvéru istým spôsobom vyhodnocovať. Práve pre tento účel máme k dispozícii rôzne metriky, určené na meranie rôznych faktorov vo vývoji softvéru. No nie každú metriku je vhodné využiť v každom projekte. Niektoré si vyžadujú isté množstvo vstupných informácií. Iné zasa nie je možné realizovať v začiatkových fázach vývoja. Metriky nám môžu pomôcť pri včasnom odhaľovaní rizík a problémov pri vývoji softvéru. Môžeme pomocou nich zistiť produktivitu a vynaložené úsilie jednotlivých členov tímu alebo jednotlivých tímov. Podľa predchádzajúcich výsledkov meraní, môžeme lepšie ohodnotiť a naplánovať ďalšie smerovanie projektu.

Pred výberom konkrétnych metrík je teda potrebné sa zamyslieť nad vlastnosťami vybraných metrík a za akým účelom ich chceme použiť.

### **Vlastnosti metrík**

Na základe určitých spoločných vlastností si môžeme stanoviť kritéria, podľa ktorých budeme jednotlivé metriky monitorovania porovnávať. Jeden z najdôležitejších dôvodov prečo samotné monitorovanie vykonávame je, aby sme zamedzili nekvalitnému ba dokonca chybnému vykonaniu niektorej fázy vývoja produktu.

Tu prichádzame teda k prvému podstatnému kritériu a ním je, ako dokáže konkrétny mechanizmus monitorovania odhaliť problém. Je tiež dôležité, že ako skoro dokáže tento problém identifikovať. V prípade, že daný problém rozpoznáme dostatočne včas, ešte sa dá problém napraviť, chybu odstrániť a fázu vývoja úspešne ukončiť. Pri metrikách, ktoré sa vyhodnocujú až po ukončení fázy vývoja, hrozí veľké riziko že sa vyskytne nejaký problém a nebude sa už dať napraviť. Preto je vhodné si v takýchto prípadoch rozčleniť fázu vývoja na menšie časti, aby sme zabránili väčším škodám.

Pri vývoji produktu zohráva dôležitú úlohu zákazník, preto treba v monitorovaní zohľadniť aj tento faktor. Každý zákazník chce mať prehľad o svojom produkte. Monitorovacie techniky umožňujú jednoduchšie prezentovať zákazníkovi jeho aktuálny stav. Môžeme mu tiež zobraziť doteraz vynaložené prostriedky. Preto je vhodné aby výstupy jednotlivých monitorovacích mechanizmov boli pre zákazníka čo najviac jednoznačné a zrozumiteľné.

Dôležitou úlohou monitorovania je odhadnúť napredovanie ďalšieho vývoja produktu. Keďže je monitorovanie úzko späté s plánovaním je vhodné, aby jednotlivé metriky boli schopné odhadovať zvyšné prostriedky potrebné na dokončenie danej úlohy. Tým sa nám uľahčí plánovanie nasledujúcich etáp vývoja. No je tu tiež riziko nesprávneho odhadu. Napríklad tento problém nastáva pri metóde, kde sa manažér jednotlivo informuje u programátorov koľko percent práce už majú dokončené a koľko ešte majú pred sebou. Programátor určí nejaký odhad, no neskôr narazia na problém a zistí, že nie je tam kde predpokladal a manažér bude musieť svoj plán znova prehodnotiť.

Závažnou vlastnosťou monitorovacích metrík je náročnosť. To znamená koľko skúseností a aké schopnosti je potrebné mať pre použitie konkrétnej metriky.

Sú dva základné prístupy pri monitorovaní a to monitorovanie ktoré sa vykonáva až po ukončení jednotlivých fáz vývoja produktu a monitorovanie počas vývoja produktu[1]. Pri variante ktorá sa uskutočňuje až po ukončení fázy vývoja vieme zákazníkovi jednoznačne prezentovať a preukázať vytvorené výsledky. Zákazník si už môže odskúšať prototypy produktu a tým dostávame dobrú spätnú väzbu. Tento prístup tiež odbreňuje manažéra od plánovania jednotlivých kontrol. Zase pri monitorovaní počas priebehu fázy vývoja má zákazník neustály prehľad o aktuálnom stave produktu. No hlavnou výhodou tohto prístupu je, že môžeme včas identifikovať problém a napraviť ho. Však nie každú metriku môžeme využiť počas vývoja, pretože v začiatkoch nemáme dostatok vstupných údajov.

## Metriky pre dĺžku zdrojového kódu

Metriky zohľadňujúce dĺžku kódu sú populárne a vyžívajú sa hlavne na zisťovanie produktivity členov tímu. Je viacero typov týchto metrík ako počet riadkov, počet tried, počet metód a podobne. Tieto metriky sú v celku jednoduché a nevyžadujú si príliš veľké skúsenosti. Pre zákazníka neposkytujú atraktívne informácie, keďže jeho skôr zaujímajú už vytvorené funkcionality a nie je preňho podstatné ako veľká časť kódu je nato potrebná. Nedajú sa využiť pri všetkých fázach vývoja keďže niektoré fázy sú skôr orientované na analýzu prípadne testovanie, kde sa neprodukuje zdrojový kód. Nie vždy zobrazujú skutočnú produktivitu programátorov. Napríklad program s rovnakou funkcionalitou môže jeden programátor napísať na 500 riadkov a ďalší využije už vytvorené knižnice a jeho kód bude mať len 200 riadkov. Ďalej sa týmto monitorovaním nezohľadňuje čas strávený pri optimalizácii a refaktorovaní kódu. Pri monitorovaní počas vývoja nám tieto metriky nemusia poskytovať relevantné výsledky, lebo nebudeme mať dostatok vstupných údajov (hlavne v začiatkoch). Z výsledkov týchto metrík nevieme určiť ako zložitý je napísaný kód.

## Komunikačná metrika

Táto metrika prezentuje dáta získané zo vzájomnej komunikácie medzi členmi tímu, vo väčších projektoch aj medzi tímami navzájom. Z jej výsledkov môžeme vidieť ako intenzívne tím pracuje na riešení úloh a tiež ktorý členovia sú ako aktívny. Táto metrika nie je veľmi náročná a nevyžaduje si veľké skúsenosti. Podľa predchádzajúcich údajov môžeme zistiť kedy a ako intenzívne máme začať pracovať, aby sme úlohu dokončili načas. A.H. Dutoia a B.Bruegge vo svojej štúdií zistili, že členovia tímu medzi sebou najviac komunikovali tesne pred dokončením etapy vývoja a naopak najmenej často v jej začiatkoch[3]. Výhodou tohto monitorovania je, že sa môže využiť v akejkoľvek časti vývoja.

## Vylepšené metriky pre dĺžku zdrojového kódu

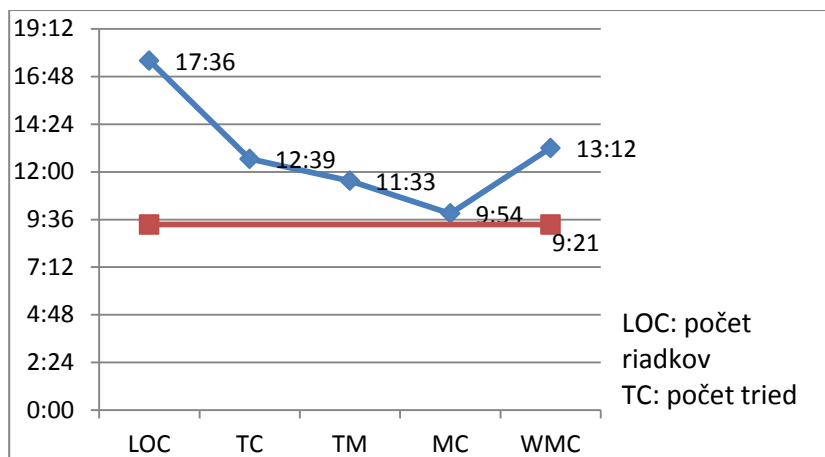
Sú to upravené metriky ktoré zobrazujú dĺžku napísaného zdrojového kódu. V týchto metrikách je navyše zohľadnený aj podiel vymazaného alebo modifikovaného kódu. Pre lepšie pochopenie môžem uviesť vzorec ktorý hodnotí počet tried:

$$MS - TC = \frac{|Classadded+Classdeleted|+Classmodified}{Total\ Class\ in\ the\ Inicial\ System} \quad (1)$$

Môžeme si všimnúť, že sa vo vzorci vyskytujú aj hodnoty upravených, vymazaných a pridaných tried.

Z toho vyplýva, že poskytujú presnejší prehľad o vynaloženej práci na vývoji. Podľa výskumu A. Wirotiakuna a P. Netisopakul sa dá pomocou nich lepšie odhadnúť čas potrebný na vývoj v ďalších etapách projektu[2]. Na obr. 1. Môžeme vidieť namerané výsledky ako ktorá metrika odhadovala potrebný čas na vykonanie úlohy. Tieto výsledky sme dostali odvodením od výsledkov, ktoré sa namerali s predchádzajúcich projektov. Vyberali sa len také projekty ktoré mali podobné funkcionality a zložitost'. Všimnite si, že

metrika zohľadňujúca celkový počet metód v triede (MC) takmer presne určila potrebný čas (9:21).



Obr. 1. Graf presnosti odhadovania potrebného času.

Tieto mechanizmy, tiež nie je vhodné používať v častiach vývoja kde neprodukuje zdrojový kód alebo pri monitorovaní počas vývoja (hlavne v začiatkoch).

## COCOMO

Táto metrika je veľmi rozšírená a používa sa hlavne na odhadnutie nákladov potrebných na vývoj projektu [4]. Názov COCOMO je odvodený zo slov „Constructive Cost Model“. Tento model má tri úrovne:

- Základná – základné hrubé odhady. Napríklad odhadneme ako dlhý bude zdrojový kód projektu.
- Stredná – pridajú sa ďalšie atribúty spojené s projektom. Projekt sa rozčlení na viacero častí.
- Podrobná – ku každej časti sa pridajú konkrétnejšie atribúty.

Metrika nie je určená na vykonávanie počas monitorovania projektu. Je ju vhodné využiť na začiatku vývoja softvéru, aby sme čo najlepšie odhadli náklady pre projekt. Môže nám napomôcť pri vykonávaní nasledovných činností:

- Rozhodovanie o investíciách – keď budeme vedieť koľko nákladov si bude projekt vyžadovať môžeme sa lepšie rozhodovať aké investície budeme vykonávať.
- Tvorba rozpočtu projektu – pred začatím vývoja je potrebné určiť si rozpočet v čom nám pomôže odhadovaná veľkosť nákladov.
- Plánovanie projektu – budeme vedieť lepšie odhadnúť kedy narábať so svojimi zdrojmi.
- Určenie ceny produktu – s presnejšieho odhadu nákladov vieme lepšie stanoviť cenu produktu.

Táto metrika je obzvlášť vhodné použiť pri konzultáciách so zákazníkom, aby presne videl začo platí a aké nákladov si to vyžaduje.

## Záver

V tejto práci som sa snažil identifikovať spoločné najdôležitejšie vlastnosti metrick monitorovania projektu. Tým som chcel objasniť podľa akých faktorov sa treba riadiť pri výbere jednotlivých metrick.

Dôležitým faktorom pri vyhodnocovaní jednotlivých metrick je obdobie v ktorom dané merania vykonávame. Pretože niektoré metriky si vyžadujú isté množstvo vstupných dát. No v prípade ak zvolíme napríklad komunikačnú metriku alebo metriku podobného typu, môžeme ju použiť kedykoľvek. Metriky zohľadňujúce zdrojový kód nám poskytujú správny obraz o našej aktuálnej produktivite no odporúčam využiť niektoré s vylepšených metrick tohto typu, keďže tie zohľadňujú viacero faktorov nie len rozsah zdrojového kódu. V začiatkovej fáze je veľmi vhodné použiť metriku pre odhad nákladov spojených s vývojom softvéru.

## Použitá literatúra

1. Tsunoda M., Matsumura T., Matsumoto K.: *Modeling Software Project Monitoring with Stakeholders*, Ninth IEEE/ACIS International Conference on Computer and Information Science, Aug. 2010, 723 - 728.
2. Wirotiakun A., Netisopakul P.: *Improving Software Maintenance Size Metrics*, International Joint Conference on Computer Science and Software Engineering (JCSSE), 2012, 334 - 339.
3. Dutoit A.H., Bruegge B.: *Communication metrics for software development*, IEEE Transactions on Software Engineering, 1998, 615 - 628.
4. Yahya, M.A.; Ahmad, R.; Sai Peck Lee; , "Effects of software process maturity on COCOMO II's effort estimation from CMMI perspective," Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on , vol., no., pp.255-262, 13-17 July 2008

## Annotation

*What metrics should I choose?*

*This work seeks to clarify the different metrics monitoring project. It also offers important features for the metrics that need to be taken into account when choosing a particular metric. I tried to identify general properties that they can be applied to all metrics. Further in the work is already specifically mentioned metrics described by these properties. For each metric are provided by its advantages and disadvantages. The aim of this work is to determine what we allow and metrics to help the reader make better choices the selection of the metrics. Because of the large number of metrics and we can all use. The combination of appropriate metrics can avoid most of the problems in software development.*