

# MERANÍM ZA KVALITNÝM KÓDOM

*Nemôžeme kontrolovať to, čo nemôžeme odmerať.*

*Pavol Fábik*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
xfabik[zavináč]stuba[.]sk

**Abstrakt.** *Pojem kvalita je nám veľmi blízky, stretávame sa s ním v živote dennodenne. Pre softvérové projekty predstavuje strategický význam. Na dosiahnutie kvality v softvérovom projekte existujú rôzne spôsoby. V eseji sa zameriavam na techniky merania zdrojového kódu, ktoré môžu vývojári použiť s cieľom zvýšenia kvality vytváraného projektu. Vďaka meraniam získajú vývojári predstavu vo forme číselnej hodnoty o kvalite ich zdrojového kódu. Túto informáciu následne môžu využiť na zlepšenie svojho kódu. Kvalitu softvéru ovplyvňuje viacero faktorov, pričom kvalita zdrojového kódu patrí medzi tie kľúčové. Jej význam je veľmi dôležitý počas vývoja softvéru, ale aj po nasadení v čase údržby.*

**Kľúčové slová:** *kvalita softvéru, meranie softvéru, kvalita zdrojového kódu*

## Úvod

Kvalita vo všeobecnosti predstavuje dosť široký pojem. Čo je to kvalitný výrobok? Prečo vravíme, že nejaký produkt je kvalitnejší ako iný produkt. V reálnom svete máme tendenciu priradovať veciam hodnoty, aby sme ich mohli porovnávať a vyhodnotiť tie najlepšie. Ale aké sú tie kritéria, ktoré používame? Príkladom môže byť automobil. Ako vieme vyhodnotiť, ktorý automobil je najkvalitnejší. Najlepšie podľa vlastností. Meraním vlastností automobilov a následným porovnaním, získame hodnoty, ktoré nám napovedia o ich kvalite a pomôžu pri rozhodovaní o ich kúpe.

Kvalita softvéru sa ohodnocuje podobne. Vyhodnocujú sa vlastnosti vytváraného softvéru, pričom musí spĺňať kritéria zo strany zákazníka a programátora. Na základe

výsledkov meraní a porovnávaní s požadovanými výkonnosťnými charakteristikami softvéru sa rozhodne, či je softvér kvalitný alebo nie. Avšak táto jednoduchá aplikácia sa dá použiť skôr na menšie projekty. Problematika vyhodnocovania kvality softvérových projektov je oveľa komplexnejšia a závisí od veľa premenných.

V eseji sa ďalej budem venovať téme, prečo je dobré merať zdrojové kódy softvérového projektu a ako nám môžu pomôcť rôzne metriky merania na zvýšenie kvality nami vytváraného softvéru.

## Meraním za kvalitou

Prečo by sme mali niečo merať? Ak chceme porovnať dve veci napríklad výšku dvoch ľudí, v prvom rade ju musíme získať. Keď máme tieto hodnoty, môžeme ich porovnávať. Ak máme dve a viac hodnôt, môžeme si z nich vyberať. Po meraní, máme možnosť výberu. Ak máme k dispozícii hodnoty, ktoré nám vyjadrujú kvalitu nášho projektu, dostávame šancu na zlepšenie týchto hodnôt a dosiahnutie zvýšenia kvality nášho projektu.

Meranie nám pomáha rozlíšiť rozdielne veci, napríklad sme schopný porovnávať, pomáha nám ohodnotiť vlastnosti projektu, napredovať a robiť správne rozhodnutia. V prvom rade potrebujeme merať, ak sme schopný merať náš softvér môžeme porovnávať, robiť dôležité rozhodnutia a vylepšovať.

### Prečo je dobré merať softvér?

Ak sa pozeráme na kvalitu z pohľadu návrhu softvéru, ako poznáme, že návrh je dobrý? Zlé návrhy sú ťažko pochopiteľné, zle sa implementujú, udržiavajú a rozvíjajú. Softvér môže fungovať správne, ale jeho návrh môže byť príliš komplexný. V etape návrhu softvéru je dôležité, ak máme viacero návrhov, použiť metriku na zistenie a porovnanie, toho, ktorý je najvhodnejší. Je lepšie použitie troch tried alebo deviatich tried?

Ak sa pozeráme na kvalitu z pohľadu kódu a pýtame sa: Je kód zložitý? Komplexné kódy sú väčšinou ťažko pochopiteľné, testovateľné, ťažko sa odlaďujú alebo menia. Ak viacej ľudí pracuje na rovnakom kóde, každý má svoj štýl programovania, aj to môže prispieť k zložitosti kódu.

Meranie je dobré hlavne na to, aby sme boli schopný ohodnotiť vlastnosti softvéru a rozhodnúť, ktorým smerom sa vydať, pričom nezáleží, v akej etape vývoja softvéru sa nachádzame. Merania je potrebné vykonávať počas celého vývoja softvéru (návrh, implementácia, testovanie, údržba, ladenie, odovzdanie).

### Aké vlastnosti môžeme meraním zachytiť

Aké vlastnosti o softvéri môžeme zistiť pomocou merania. Vlastnosti softvéru predstavujú jeho kvalitu [1]:

- Pokúšame sa zachytiť kvalitu softvéru, aký dobrý je náš produkt,
- úsilie, odhad zdrojov na vývoj softvéru,
- pokrok vo vývoji (ako ďaleko sme, koľko ešte ostáva),
- komplexnosť a zložitosť úlohy,
- kvalita procesu, aký dobrý je proces vývoja softvérového produktu,

- faktory kvality: použiteľnosť, správnosť, efektívnosť, modulárnosť, prenositeľnosť, jednoduchosť, zrozumiteľnosť, bezpečnosť, robustnosť, kompletnosť.

## Meraním ku kvalitnému kódu

Celková kvalita vytváraného softvéru môže byť rozdelená do viacerých aspektov. Napríklad:

- správnosť operácií,
- požadovaná funkcionálnosť,
- jednoduchosť používania,
- bezpečnosť pred hrozbami.

Kvalita zdrojového kódu predstavuje ďalší aspekt. Môže byť definovaná ako tie aspekty zdrojového kódu, ktoré nemajú vplyv na správanie bežiackej aplikácie, avšak majú vplyv na schopnosti vývojárov porozumieť softvéru a jednoducho aplikovať zmeny v budúcnosti.

## Čo merať v softvéri

Čo merať závisí od nášho cieľa. Čo chceme dosiahnuť našim meraním. Nasledujúci zoznam prezentuje príklady meraní v softvéri [2]:

- Veľkosť softvéru, rôzne spôsoby merania veľkosti,
- viazanosť komponentov v module, chceme dosiahnuť vysokú previazanosť kódu, nechceme vložiť nesúvisiace funkcie, metódy, triedy do rovnakého modulu,
- závislosti modulov medzi sebou, chceme vedieť ako veľmi sú module previazané, závislé medzi sebou,
- zložitosť kódu alebo organizácia kódu,
- objavená chybovosť za jednotku počtu riadkov kódu, napríklad 1 chyba každých 5 riadkov, predstavuje vysokú chybovosť, toto meranie neberie do úvahy dopad chyby na softvér,
- čas strávený na rozdielnych procesoch,
- počet požiadaviek na softvér, ak máme veľa príkladov použitia, získavame predstavu o dĺžke vývoja softvéru.

Ak chceme niečo merať, v prvom rade potrebuje získať dáta na meranie. Dáta môžeme získať automaticky aj manuálne. Pre rýchlosť vývoja je najlepšie používať pomocné nástroje, ktoré automaticky získajú informácie o softvéri urobia porovnanie, informujú nás o pokroku a upozornia nás na problémy.

## Metriky, ktoré môžu pomôcť na zvýšenie kvality kódu

### LOC

Metrika počtu riadkov zdrojového kódu je asi najzákladnejšia metrika na meranie obsahu zdrojového kódu. Je orientovaná na objem a slúži k odhadu veľkosti softvéru. Je ľahko merateľná a predstavuje súčet počtu riadkov kódu programu. Existuje niekoľko variant tejto metódy, príklad variant podľa Jonesa [3]:

1. sčítanie iba vykonateľných riadkov,
2. sčítanie vykonateľných riadkov, plus definovanie dátových typov,
3. sčítanie vykonateľných riadkov, plus definovanie dátových typov a komentáre,
4. sčítanie vykonateľných riadkov, definovanie dátových typov, komentáre, jazyk kontroly práce (napr. ošetrovanie výnimiek apod.),
5. spočítanie riadkov ako fyzických riadkov na výstupe obrazovky,
6. spočítanie riadkov ukončených logickým oddeľovačom.

Táto metrika nám hovorí o veľkosti programov, môže nám aj povedať, či sú dané kódy dobré? Aký zložitý je kód? Môžeme povedať, že program s 1000 riadkami je vysoko kvalitný, pričom program s 500 riadkami je málo kvalitný? Nie, pomocou tejto metriky nedokážeme spraviť takéto závery. LOC nám iba vraví že program s 1000 riadmi je rozsiahlejší ako program s 500 riadkami.

Prečo by sme mali teda použiť túto metriku. Je vhodná na predvídanie veľkosti úsilia potrebného na vývoj programu, ako aj na stanovenie produktivity vývoja a udržateľnosti ak je softvér nasadení. Riadky kódu sú užitočné, ale neposkytujú všetky faktory kvality.

### Halsteadove metriky

Halstead vytvoril zoznam metrick na meranie zložitosti modulu programu priamo zo zdrojového kódu s dôrazom na výpočtovú zložitosť. Halsteadova metrika počíta dĺžku a veľkosť softvéru na základe počtu operátorov a operandov. Na základe jednoduchých meraní Halstead vynašiel systém rovníc, ktoré vyjadrujú celkovú dĺžku programu, odhadovaný minimálny objem pre algoritmus, skutočný objem, úroveň programovania (metriky zložitosti softvéru), programovú náročnosť, a ďalšie rovnice, napr. pre programátorské úsilie a predpokladaný počet chýb v softvéri. Programové úsilie predstavuje úsilie investované v človekomesiacoch do programovania modulu. Použitím týchto rovníc môžeme kvantifikovať kvalitu nášho zdrojového kódu.

### Cyklomatická zložitosť

Cyklomatická zložitosť je metrika, ktorá bola predstavená pánom McCabom na meranie množstva rozhodovacej logiky v samostatnom softvérovom module. Tento systém meria počet nezávislých ciest v programe, čo udáva číselnú hodnotu zložitosti. V praxi je to súčet čísel testovacích podmienok programu.

Podľa skúseností niektorých vývojárov, cyklomatická zložitosť je spojená s existenciou chýb v kóde. Ak máme triedu s vysokým číslom cyklomatickej zložitosti, určite obsahuje chyby. Nezáleží na veľkosti projektu, či sa jedná o malý alebo veľký

projekt. Metóda ktorá zobrazí veľké číslo cyklomatickej zložitosti predstavuje miesto, na ktorom by mal programátor ešte pracovať. Ak sa tam chyby nevyskytujú, môžu sa tam v budúcnosti objaviť, pri zmenách v iných častiach kódu, na ktoré je táto časť naviazaná. Vhodné použitie na vyhodnotenie, kedy robiť prehliadku kódu. Ak vyjde vysoké číslo, mali by ste sa zamyslieť nad refaktoringom a prehliadkou kódu. Obrovskou výhodou cyklomatickej zložitosti, i keď si to možno väčšina vývojárov neuvedomuje je, že nám poskytuje počet koľko testov musíme napísať, aby sme kompletne otestovali všetky vetvy v rámci metódy. Ponúka nám rýchlu predstavu koľko času a úsilia, bude treba na doplnenie testov k existujúcemu kódu. Ak zložitost' je 20 viem, že budem musieť napísať prinajmenšom 20 testov. Ak sa vývojári počas vývoja softvéru ponáhľajú a nestíhajú plne otestovať svoj kód, číslo zložitosti raketovo zväčšuje. Po vydaní produktu to číslo ide automaticky dole, pretože programátori majú čas na refaktoring kódu.

Nepoužívaním Cyklomatickej zložitosti pri vývoji sa vývojári oberajú o výkonný nástroj, ktorý im napomáha zvýšiť kvalitu vytváraného kódu. Menej chýb vyššia kvalita.

## Záver

Mať kvalitný zdrojový kód neznamena automaticky mať kvalitný softvér. Na kvalitu softvéru vplýva veľké množstvo aspektov a toto je len jeden z nich. Avšak svojou dôležitosťou, predstavuje jednu z rozhodujúcich vlastností a je nevyhnutné ju neustále vylepšovať. Stav zdrojového kódu majú na svedomí samotní vývojári. Je to zrkadlo ich práce a preto je v ich záujme používať techniky, napríklad aj tie spomenuté v tejto práci, aby vedeli ohodnotiť svoje zámery a neustále napredovali vpred.

I keď tieto metriky vznikli v pomerne informatickom stredoveku, čas ukázal, že aj teraz majú stále čo ponúknuť a na obdobie svojho vzniku boli nadčasové. 1

## Použitá literatúra

1. Mills, Everaldo E.: *Software Metrics*, [Online], 1988, [Dátum: 5. 10. 2011] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.6110&rep=rep1&type=pdf>
2. Sarda N.L.: *Software Metrics and Quality*, [Online], 2008, [Dátum: 28. 11. 2011] <http://www.youtube.com/watch?v=KqDIDubS-OU>
3. Kan, S.H.: *Metrics and Models in Software Quality Engineering*, 1995.

## Annotation

### *Using measure towards code quality*

*Quality is very familiar to us. We are in contact with it on everyday basis. It has strategic meaning for the software projects. There are several techniques how to achieve quality in software projects. I will analyze source code measuring techniques in this essay, which can developers use to increase quality of their projects. Measures can give developers exact number, which can help them in further coding and increasing quality. There are several quality factors which affect software quality*

**6** *Pavol Fábik*

*and source code quality is one of the most crucial. Its meaning is very important during development and in maintenance*