

MERANÍM K LEPŠIEMU SOFTVÉRU ALEBO AKÉ SÚ IDEÁLNE MIERY SOFTVÉRU?

*Nie je jednoduché odmerať niečo, čo je tak abstraktné
ako softvér. A už vôbec nie je ľahké zistiť, čo
nameraná hodnota znamená.*

Michal Granec

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

Abstrakt. Každý produkt by sa mal snažiť, čo najlepšie splniť požiadavky naňho kladené a byť čo najkvalitnejší, ako to len pôjde. Softvérový manažér kvality musí čeliť viacerým problémom, aby dosiahol úroveň kvality softvéru takú, aká je od neho požadovaná. Jedným zo spôsobov, ktorý manažér kvality používa na kontrolu kvality, je použitie softvérových metrík. Tie nám môžu pomôcť odmerať dôležité rozmery softvéru a získať o ňom hmatateľnejšiu predstavu a to aj napriek jeho vnútornej komplexnosti. Používanie metrík avšak nie je absolútne riešenie a musíme si dať pri tejto technike pozor na rôzne nástrahy, ktoré nás môžu prekvapiť. V tejto eseji analyzujem ich všeobecné uplatnenie, ale aj najčastejšie problémy, na ktoré narazíme pri ich používaní. Tieto problémy ďalej rozoberám a pokúšam sa prísť na riešenie týchto problémov.

Kľúčové slová: manažment kvality softvéru, softvérové metriky, vlastnosti softvérového produktu

Úvod

Proces vývoja softvéru prechádza rôznymi štádiami. Na rozdiel od ostatných inžinierskych odvetví, sa pri vyvíjaní softvéru stretávame s fenoménom neviditeľnosti softvéru. Počas všetkých štádií tvorby vzniká produkt, ktorý nie je hmatateľný a o ktorom nevieme povedať, či je alebo nie je funkčný a tým sa vynára problém určenia kvality

daného softvérového produktu. Ako povedal Tom DeMarco vo svojom diele [1]: „Nemôžete kontrolovať to, čo neviete odmerať.“ Tým chcel naznačiť, že nemôžeme zaručiť, že proces a produkt budú spĺňať požiadavky na nich kladené, pokiaľ nebudeme mať nejaké informácie, ktoré nám potvrdia správnosť našich odhadov. Táto skutočnosť rezonovala ešte hlasnejšie v období softvérovej krízy. Jednou z odpovedí bol rozvoj softvérového manažmentu kvality. Manažment kvality sa snaží kontrolovať proces vývoja softvérového produktu, lebo len vďaka kontrolovanému procesu vývoja môžeme zaručiť, že výstupný produkt bude kvalitný. Manažér kvality disponuje mnohými technikami a metódami, ktoré využíva na kontrolu kvality, medzi inými aj technikou merania softvéru.

Meranie softvéru

Ako sa teda môžeme dozvedieť aspoň o nejakej úrovni kvality nášho produktu počas jeho vývoja? Ako som spomínal, jednou z metód a techník, ktoré sú používané v manažmente kvality, je aj mechanizmus merania softvéru a s tým spojené softvérové metriky. Práve vďaka metrikám sa môžeme dozvedieť užitočné informácie už počas vývoja produktu, a to vďaka skutočnosti, že počas fázy vývoja máme okrem dokumentov analýzy a návrhu dostupný zdrojový kód produktu. Ten dostatočne vypovedá o funkcionalite a v konečnom dôsledku aj o kvalite celého produktu. Tým nechcem povedať, že metriky sa môžu používať len počas vývoja produktu. Práve naopak, som toho názoru, že metriky sú veľmi vďačným zrkadlom, ktoré môžeme nastaviť nášmu produktu počas všetkých fáz vývoja, a nielen to, ale pomocou softvérových metrik sme schopný zmerať aj úroveň kvality vývojového procesu, ktorá je rovnako dôležitá ako úroveň kvality produktu.

Všeobecne delíme metriky na produktové a procesné. Produktové merajú produkt počas všetkých fáz vývoja a môžu merať rôzne vlastnosti produktu od zložitosti softvérového návrhu, cez veľkosť finálneho programu, až po počet strán dokumentácie. Na druhú stranu, procesné metriky môžu merať vývoj softvérového procesu z hľadiska tímu alebo organizácie, ako napríklad strávený vývojový čas, typ použitej metodológie alebo úroveň zručností programátorského tímu.

Komplexnosť dnešných systémov

Komplexnosť dnešných systémov by sa dala prirovnať k počasiu. Každý vie povedať, aké je dnes počasie a či bolo lepšie ako včera. Pokiaľ sa ľudí spýtame, prečo tomu tak je, väčšina nám bude podsúvať odpovede ako teplota vzduchu, stav oblačnosti, sila a smer vetra alebo tí zdatnejší nám povedia o vlhkosti vzduchu a o iných parametroch. Jedným slovom nám budú odpovedať vo forme metrik, ktoré nám majú vypovedať o celkovom stave a pokiaľ sme aspoň trochu zdatní v oblasti meteorológie, tak si môžeme utvoriť presnú predstavu, aké to počasie dnes je. Na rovnakom princípe fungujú aj softvérové metriky. Vypovedajú nám o jednotlivých rozmeroch produktov a pokiaľ ich vieme dobre interpretovať, dokážeme vidieť dôležité prvky produktu aj skrz jeho komplexnosť. Zdanlivo to nie je také jednoduché, ako to znie, a to z dôvodu obrovského množstva vyvinutých metrik, nedostatkom ich teoretického základu, štatisticky nepostačujúcim experimentálnym overením a problémom s ich interpretáciou [2].

Ani v dnešnej dobe neexistuje metrika, ktorej výstupná hodnota by vypovedala o komplexnosti celého softvéru. Určite existuje mnoho takých, ktoré dokážu merať určitú úroveň komplexnosti, ale sú použiteľné za veľmi obmedzujúcich podmienok. Jednou z možných príčin tejto situácie, môže byť viacnásobný rozmer softvéru. Nedokážeme navrhnúť metriku, ktorá by bola schopná zmerať každý jeden rozmer softvéru a ešte ich aj vzájomne porovnávala, aby nakoniec dospela k jednému výsledku. Problémom by podľa mňa mohla byť aj skutočnosť, že ak si vyberieme dva rôzne produkty na rovnakej úrovni kvality, tak sa môžu natoľko líšiť vo svojich rozmeroch, že naša absolútna metrika by zlyhala, lebo by nedokázala tieto odlišnosti reálne zahrnúť. Ako príklad môžem uviesť jednoduchú metriku na meranie dĺžky kódu (LOC). Pri rovnakom rozmere, by programovacie jazyky vyššej úrovni mali podstatne kratší kód ako programovacie jazyky na nižšej úrovni. Rovnako by zostala neznáma výpovedná hodnota takejto metriky. Čo nám táto absolútna metrika vraví? Pokiaľ by som nevedel, čo všetko merala a ako vnútorne tieto hodnoty vzájomne porovnávala, tak by som nebol schopný interpretovať výslednú hodnotu tejto metriky. Preto som toho názoru, že je strata času snažiť sa objaviť jednu absolútnu metriku, ktorá by merala každý rozmer softvéru. Lepším spôsob sa mi javí použitie súboru metrík, ktoré merajú pre nás dôležité rozmery softvéru a majú pre nás určitú výpovednú hodnotu, s ktorou vieme ďalej pracovať. Otáznym zostáva, aké rozmery sú pre nás, ako manažéra kvality dôležité.

Čo všetko chceme vlastne merať? Problém komplexnosti dnešných systémov, len sťažil proces identifikácie pre nás potrebných rozmerov softvéru a tým aj problém dodržania a určovania kvality softvérového systému. Túto skutočnosť ešte umocňuje aj veľká abstraktnosť spojená so softvérom. Samotné miery softvéru nie sú fyzicky hmatateľné a nie je to niečo, čo je normálnym okom vidieť. Ako ideálny príklad by som mohol uviesť miery pohára. Pokiaľ zistím, že objem pohára je pol litra, tak si každý človek utvorí predstavu o približnej veľkosti pohára alebo aspoň množstve tekutiny, ktorý je schopný obsiahnuť. Rovnako by sme nemali problém s výškou pohára a jeho polomerom, ale už by mohli nastať menšie problémy s interpretáciou týchto veličín. Pri softvérových mierach je situácia zložitejšia. Pri ich interpretácii nám často chýba nejaký návod, ako vyhodnotiť výsledky, ktoré dostaneme. Práve nedostatok teoretických základov a štatisticky nedostatočne potvrdených výsledkov sa môžu prejaviť v zlom interpretovaní výsledkov [2].

Čo nám povie McCabova cyklomatická zložitost', pokiaľ nám vráti výsledok osem? Je to akurát, veľa alebo málo? Sám som zostal zaskočený, koľko rôznych názorov som počas môjho surfovania na Internete našiel. Hlavne na diskusných fórach som sa stretol s diametrálne odlišnými názormi. Dokonca som sa stretol s názorom, kedy dotyčnému prišlo zbytočne zložitú sa s touto metriku zaoberať. Nevie, či dotyčný považuje túto metriku za zložitú, pretože jej nerozumie alebo len nevie využiť výstupnú hodnotu tejto metriky. Ja osobne považujem túto metriku za veľmi užitočnú. Ona samotná nám môže veľa povedať o lineárne nezávislých cestách, ktoré vedú skrz zdrojový kód softvéru. Jej výsledok sa dá napríklad využiť tak, že každá takáto cesta by mala byť pokrytá testom. Je naozaj otáznne, aká by mala byť ideálna hodnota tejto metriky, ale s týmto sa podľa mňa, musí popasovať manažér kvality, ktorý musí byť dostatočne kvalifikovaný a musí dostatočne poznať všetky faktory, ktoré by mal brať do úvahy pri interpretovaní výsledkov jednotlivých meraní.

Som presvedčený, že riešením spomínaných problémov spočíva v nájdení postupu, ktorý by nám pomohol s výberom správnych metrík a rozhodnutím sa, čo vlastne chceme merať. Niektoré problémy, ako ich nedostatočný teoretický základ a štatisticky nepostačujúce experimentálne overenie nie sme schopní ovplyvniť. Problém výberu správnych metrík sa nám aspoň trochu snaží uľahčiť rámec cieľ – otázka – metrika (angl. goal question metric) [3].

Ktoré metriky si vybrať?

Tento cieľovo orientovaný rámec pre softvérové metriky vyvinul Dr. Victor Basili a pristupuje k celému problému zhora dole. Hlavnou myšlienkou je, aby si vývojový tím vytvoril projektové ciele a na tie potom aplikoval sériu krokov, ktoré zabezpečia ich splnenie. Ciele sa postupne rozbiehajú na otázky a na tie sa vyberú vhodné metriky, ktoré majú zaručiť, že budú zodpovedané vo forme merania. Následne sa zozbierajú dáta, overia sa a vývojový tím zistí, či splnil danú otázku a teda časť cieľa. O výsledkoch má vývojový tím informovať zákazníka. Práve posledný krok považuje autor Victor Basili za kľúčový, keďže podľa neho je to jeden z hlavných dôvodov predchádzajúcich krokov [3].

Tu si dovoľím čiastočne nesúhlasiť s tvrdením pána Basiliho. Podľa môjho názoru je určite potrebná informácia klienta o stave produktu, za ktorý si zaplatil nemalé peniaze, ale určite nie je vhodné mu rozprávať o každom vykonanom meraní. Veď načo, by mu bola informácia o hĺbke stromu dedičnosti. Pokiaľ by nebol zákazník aspoň trochu infromaticky zdatný, tak táto informácia ho môže prinajhoršom vystrašiť a môže sa cítiť zmäteno, prečo mu podsúvate takéto informácie. Na druhú stranu, informácie o čase spustenia programu, jeho veľkosti alebo celkovom čase vývoja produktu sa môžu prinajlepšom stretnúť s nadšením zo strany zákazníka alebo aspoň kladnou odozvou. Táto skutočnosť sa môže zísť hlavne malým softvérovým firmám, kde udržanie zákazníka znamená pre nich existenčný problém. Zákazník, ktorý má pocit, že je o jeho produkt dobre postarané a je vyvíjaný s maximálnym nasadením, nemá dôvod sa v budúcnosti obrátiť s novou zákazkou na inú firmu.

Hlavný prínos tohto rámca vidím najmä v tom, že pomáha tímu si ujasniť, ktoré rozmery produktu sú pre nich dôležité a postupovať tak, aby boli splnené. Je v tom vidieť určitú podobnosť s technikou rozdeľ a panuj (lat. per partes), kedy celok je rozdelený na menšie časti, ktoré môžu byť ďalej niekoľko krát rozdelené na ešte menšie komponenty, na ktoré už dokážeme nájsť riešenie, čo sa prejaví v nasadení správnej metriky. Tento rámec môže byť užitočný aj vo veľkej firme, kde by mohli byť jednotlivé ciele pridelené rôznym tímom, ale aj v malom tíme, akým je náš tímový projekt. V ňom si viem predstaviť, že by sa celý tím zúčastnil procesu tvorby cieľov aj otázok s manažérom kvality, a potom by sa zodpovednosť za splnenie jednotlivých cieľov a otázok rozdelilo medzi členov. Týmto spôsobom by mal tím ujasnené, aké rozmery softvérového produktu sú pre nich dôležité a zároveň, aké metriky budú potrebovať na meranie softvéru.

Metriky sú dobrý sluha, ale zlý pán

Metriky nám vždy môžu veľmi dobre poslúžiť, no všetko treba s mierou. I keď som pred chvíľou tvrdil, že metriky nám pomáhajú vypovedať o kvalite produktu a celého procesu

vývoja, myslím si, že by to nemala byť modla každého programátora a analytika. Kaner vo svojom diele považuje honbu za čo najlepšimi číslami metrík za značné riziko pri softvérovom vývoji, ktoré niekedy môže byť viac škodlivé, ako prospešné [4]. Sám hovorí, že ľudia dokážu byť veľmi vynaliezaví, hlavne pokiaľ existuje niečo, voči čomu ich môžeme odmerať a ohodnotiť. Napríklad pokiaľ chceme viacej chýb, tak ich dostaneme alebo pokiaľ chceme peknú zaoblenú krivku, tak nám ju ľudia vyrobia. Stávajú sa dokonca prípady, kedy zle nastavené spôsoby merania viedli vo firme alebo v tíme k podvádzaniu, keďže zamestnanec sa snažil vylepšiť si svoje produkčné čísla a vyvíjal nadštandardne dlhý a neefektívny kód, čo určite nie je to, za čo si zákazník zaplatil.

Rozhodujúcim faktorom pri interpretácii výsledkov meraní je práve ľudský faktor v podaní manažéra kvalita. Ten ovplyvní, aký veľký dopad budú mať výsledky metrík na celkový vývoj softvéru. Práve jeho kvalifikácia a znalosť produktu alebo procesu sú kľúčové pri vyhodnocovaní jednotlivých výsledkov meraní. Sám musí vedieť, či výsledok metriky je adekvátny pre daný produkt a či vyvodíť určité kroky na zlepšenie výsledkov. Myslím si, že manažér kvality môže získať podstatnú časť svojich skúseností práve z konkrétnych projektov, na ktorých sa zúčastní. Hlavne z dôvodu, že existuje nepreberné množstvo rôznych softvérových produktov, ktoré sa môžu diametrálne odlišovať a čím bude mať manažér kvality viac príležitostí sa dostať k takýmto rôznym produktom, tým lepšie bude zvládať pochopiť jednotlivé rozdiely medzi týmito produktmi a tie následne aplikovať pri používaní metrík. Tiež sa nedajú vždy aplikovať rovnaké metriky a nie vždy sa môžu porovnávať výsledky rovnakých metrík aplikované na rozdielne produkty.

Jedným z najväčších prínosov softvérových metrík vidím práve v skutočnosti, že dovoľujú porovnávanie dvoch podobných produktov. Ako som si sám všimol, existuje veľké množstvo vedeckých odborných prác, v ktorých sa vymýšľajú nové aplikácie, spôsoby, metodiky alebo techniky. Autori týchto prác veria, že práve ich práca prináša niečo nové, čo je efektnejšie, rýchlejšie a všeobecne lepšie, než všetko čo bolo pred nimi. Ako argument často používajú, okrem iného, aj softvérové metriky, pomocou ktorých chcú prakticky a hlavne hmatateľne čitateľovi ukázať, prečo majú pravdu. Otázne je, či skutočne môžeme bez zábran porovnávať produkty práve na základe metrík? Odpoveď je áno, ale aj nie. Tá nerozhodnosť mojej odpovede pramení zo skutočnosti, že jednotlivé softvérové produkty sa môžu veľmi líšiť vo svojich charakteristikách a práve v tomto prípade by bolo použitie metrík, ako spôsob dôkazu rozporuplné, lebo by sme v podstate porovnávali neporovnateľné. Tým nechcem tvrdiť, že neexistujú metriky, ktoré sú dostatočne univerzálne, a ktoré sú vo väčšine prípadov použiteľné. Problémom ale je, že treba vždycky zväziť rozdielnosti týchto produktov a mať ich na pamäti pri meraní. Iba v takom prípade môžeme porovnávať dva produkty.

Záver

V eseji som sa zaoberal spôsobmi merania softvéru a metrikami, ktoré sú s tým spojené. Rozoberal som najčastejšie problémy spojené s ich použitím, medzi ktoré určite patrí problém výberu správnej metriky. Taktiež som sa zaoberal pojmami abstraktnosti a neviditeľnosti softvéru, ktoré spolu s komplexnosťou dnešných systémom prispievajú k obťažnejšiemu určaniu kvality produktu a tiež sťažujú identifikáciu potrebných rozmerov softvéru pre vývojový tím. Za možnú odpoveď na túto otázku považujem rámec

cieľ – otázka – metrika, ktorá nám môže pomôcť vybrať správnu metriku. Som toho názoru, že tento rámeček môže byť veľmi užitočný a jeho nasadenie by som odporúčal aj v malom tíme, ktorý má často problém si uvedomiť, aké rozmery a parametre softvéru sú pre nich dôležité. V eseji bol tiež spomenutý problém správneho použitia metrik, pretože niekedy môže byť jeho použitie až kontraproduktívne. Riešenie tohto problému vidím len a len na pleciah manažéra kvality, ktorý musí správne nastaviť nasadenie metrik vo firme alebo tímu a zároveň musí správne interpretovať výsledky jednotlivých meraní. Problémov spojených s použitím metrik je podstatne viac, no pokiaľ by som sa im chcel všetkým venovať, značne by som presiahol rozsah tejto práce. Do budúcnosti by určite mohlo byť zaujímavé rozobratie ďalších problémov spojených s meraním softvéru. Metriky aj napriek tomu považujem za výborný a užitočný nástroj manažmentu kvality, ktorý sa treba naučiť správne používať a ktorý môže mať veľkú výpovednú hodnotu o stavu kvality softvérového produktu alebo procesu.

Použitá literatúra

1. Demarco, T.: Controlling Software Projects: Management, Measurement, and Estimates. Upper Saddle River (USA) : Prentice Hall PTR, 1986.
2. Everaldo, E., M.: Metrics in the software engineering curriculum. In: *Annals of Software Engineering*. Red Bank (USA) : J. C. Baltzer AG, Science Publishers, roč. 6, 1999, č. 1-4, s. 181-200.
3. Basili, V., Caldiera, G., Rombach, H. D.: Goal Question Metric Approach. In: *Encyclopedia of Software Engineering*. John Wiley & Sons, Inc., 1994, s. 528-532.
4. Kaner, C., Bond, W. P.: Software Engineering Metrics: What Do They Measure and How Do We Know?. In: *METRICS 2004*. IEEE CS, roč. 8, 2004, s. 1-12.

Annotation

Measuring to better software or what are the ideal measures of software?

Every software product should try to meet the demands that are required from it and it should be as close to these requirements as it can. Software quality manager has to face several problems to reach such level of quality of product that is required. One of the ways which quality manager uses for controlling quality is use of software metrics. They can help us to measure important dimensions of the software and get more tangible idea of it, despite its internal complexity. Using software metrics is not absolute solution and we must be careful with this technique due to variety of pitfalls which can surprise us. In this paper, I discuss the general application of software metrics, but also the most common problems encountered during their use. I further discuss these issues and try to figure out a solution to these problems.