

AKO VHODNE KOMBINOVAŤ SOFTVÉROVÉ METRIKY?

Keď jedna metrika nestačí...

Matúš Hitka

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
mhitka@gmail.com

Abstrakt. Meranie v softvérovom projekte je najlepším spôsobom na určenie stavu projektu počas jeho realizácie. Na meranie v softvérovom projekte slúžia rôzne metriky. Avšak výpovedná hodnota jednotlivých metrik môže byť skreslená najmä efektom, že samotné meranie má vplyv na psychiku človeka. Môže to viesť až k takým extrémom, že sa vývoj produktu začne orientovať na dosiahnutie lepších výsledkov merania a nie na dosiahnutie určitého cieľa. V tejto eseji rozoberám vhodnosť a použiteľnosť troch často používaných softvérových metrik: počet riadkov kódu (LOC), funkčné body a hustota chýb. Každá z týchto metrik má svoje výhody, ale i nevýhody. Ani jedna z nich však nepokrýva všetky etapy vývoja softvérového produktu. Preto je potrebné ich vhodne skombinovať.

Kľúčové slová: meranie v softvérovom projekte, metrika, počet riadkov kódu (LOC), funkčné body, hustota chýb

Úvod

Monitorovanie projektu slúži nie len na určenie stavu projektu počas jeho realizácie, ale i na skvalitnenie riadenia. Nezameriava sa len na kontrolu vykonanej práce či splnenie cieľov. Vďaka monitorovaniu sa môžu včas identifikovať prichádzajúce problémy a prijať potrebné opatrenia, aby negatívne vplyvy na priebeh projektu boli čo najmenšie.

Stav projektu je možné určiť jednoducho tak, že sa porovná súčasný stav s predchádzajúcim stavom. Pri softvérovom projekte to však nie je jednoduché, pretože niekedy je problematické:

2 Matúš Hitka

- určiť charakteristiky, ktoré reprezentujú jeho stav
- správne vyhodnotiť tieto charakteristiky

Na meranie v softvérovom projekte sa používajú *softvérové metriky*. Medzi základné a často používané metriky patrí: počet riadkov kódu, metóda funkčných bodov či hustota chýb, ale aj mnoho ďalších.

Zároveň však treba mať na mysli, že samotné meranie môže negatívne vplyvať na psychiku človeka, ktorý sa začne sústrediť na dosiahnutie lepších výsledkov merania a nie na dosiahnutie cieľa. Výpovedná hodnota metriky teda nemusí odrážať skutočný stav vykonanej práce.

Vybrané metriky

Na pokrytie celého procesu vývoja softvérového produktu som sa rozhodol použiť 3 hlavné metriky, ktoré podrobnejšie rozoberám:

- počet riadkov kódu
- funkčné body
- hustota chýb

Počet riadkov kódu

Obľúbenou a veľmi často používanou metrikou je počet riadkov kódu (LOC). Používa sa najmä na sledovanie vykonanej práce a produktivity, ale taktiež je pomocou nej možné odhadnúť čas, ktorý je potrebný na dokončenie projektu.

C. Jones vo svojej práci [2] tvrdí, že je hazardné používať metriku, ktorá penalizuje vysoko úrovňové programovacie jazyky, čo je prípadom metriky LOC. Cena jedného riadku kódu vo vysoko úrovňovom jazyku je podstatne nižšia ako cena riadku v nižšom jazyku napriek tomu, že s produktivitou práce je to naopak. Vo vyššom jazyku vývoj aplikácie zaberie podstatne menej času ako v nižšom jazyku.

Nevhodné použitie metriky LOC môže taktiež viesť programátora k tomu, že začne písať „riedke“ kódy, aby dosiahol dobré výsledky merania. Prestane sa sústrediť na to, ako čo najefektívnejšie dosiahnuť splnenie úlohy.

Ďalším negatívnym ukazovateľom tejto metriky je, že niekedy nehovorí nič o vykonanej práci. Pri optimalizovaní programu sa môže stať, že sa viac riadkov programu zmaže ako pridá. Ak by sa produktivita práce merala výlučne len počtom riadkov kódu, programátor by mal za dané obdobie optimalizácie nulovú produktivitu, pričom hodnota jeho práce vzhľadom na vyvíjaný produkt môže byť veľmi vysoká. Preto je potrebné pri tejto metrike zaznamenávať aj popis vykonanej práce a pri vyhodnocovaní brať tento ukazovateľ do úvahy.

Osobne si myslím, že používanie metriky LOC ako smerodajného ukazovateľa na monitorovanie projektu nie je najvhodnejšie. Avšak úplne by som túto metriku nezavrhol. Jej hlavnou výhodou je jej jednoduchosť a možnosť automatického sledovania. Ak sa rozumne vyhodnotia jej výsledky, je možné veľmi ľahko sledovať množstvo vykonanej práce, ale aj produktivitu jednotlivých programátorov.

Funkčné body

Na určenie veľkosti systému sa často používa metóda tzv. „funkčných bodov“ (anglicky function points - FP). Vychádza priamo zo špecifikácie, kde sa určí počet charakteristík produktu a miera ich zložitosti. Výsledná hodnota sa určí nasledovne:

$$|FP| = \sum (\text{pocet_charakteristik} \times \text{zlozitosť_charakteristiky})$$

Význam funkčných bodov spočíva najmä v tom, že ich počet sa nemení pri použití rozličných programovacích jazykov. Programy napísané vo vyšších jazykoch majú rovnaký počet funkčných bodov ako programy v nižších jazykoch, keďže tieto body sa určujú zo špecifikácie požiadaviek a výsledná funkcionálna produktivita je rovnaká.

Pomocou metódy funkčných bodov je možné určiť produktivitu práce (meranú ako počet implementovaných funkčných bodov za mesiac), ale aj cenu produktu (cena jedného funkčného bodu). Na základe priebežných výsledkov týchto meraní sa môžu presnejšie určiť odhady času potrebného na dokončenie produktu, ako aj ceny výsledného produktu.

Funkčné body nie sú dokonalou metrikou, ale sú veľmi užitočné [1]. Podľa mňa ich hlavný význam spočíva v tom, že pomáhajú projektovým manažérom určiť veľkosť a zložitosť projektu ešte pred samotnou implementáciou, keďže sa určujú priamo zo špecifikácie. Taktiež je manažér schopný pomocou nich lepšie vysvetliť zákazníkovi dopad zmeny požiadavky na systém v priebehu vývoja projektu.

Miernou nevýhodou tejto metriky je obtiažnosť určenia charakteristík produktu a subjektívne určenie miery zložitosti týchto charakteristík. Tieto skúsenosti musí nadobudnúť manažér časom a praxou. Ak však tento hendikep prekoná, použitie funkčných bodov prináša veľmi užitočné výsledky.

Chybovosť

Chybovosť produktu pokrýva fázu testovania a nasadenia aplikácie, resp. jednotlivých modulov, verzií a pod. Často sa uvádza ako hustota chýb, ktorá je určená pomerom zistených chýb produktu vzhľadom na veľkosť produktu. Veľkosť produktu býva zvyčajne určená pomocou metriky LOC alebo FP.

$$\text{hustota chýb} = \frac{\text{pocet chýb}}{\text{velkosť produktu}}$$

C. Jones [2] zastáva názor, že použitie tejto metriky nie je celkom vhodné. Svoje tvrdenie však argumentuje na príkladoch, v ktorých sú použité rozličné programovacie jazyky. Podľa mňa však pri použití konzistentného prostredia táto metrika prináša užitočné informácie. Je možné pomocou nej sledovať hustotu výskytu chýb v jednotlivých verziách programu.

Návrh kombinácie jednotlivých metrík

Je zrejmé, že pre uspokojenie potrieb na meranie softvérového projektu nebude stačiť použitie jednej metriky. Každá metrika má svoje pre aj proti. Ďalším faktom je, že ani

4 Matúš Hitka

jedna metrika nepokrýva všetky etapy vývoja softvéru. Preto navrhujem používať nasledovnú kombináciu spomínaných metrík.

Na odhad veľkosti produktu ešte pred samotnou implementáciou je vhodné použiť metriku funkčných bodov. Vychádza priamo zo špecifikácie a pomocou nej je možné naplánovať priebeh jednotlivých etáp implementácie, odhadnúť čas potrebný na vývoj projektu, ako aj jeho cenu.

Pri projektoch malých až stredných rozmerov je väčšinou použité jednotné implementačné prostredie. Preto je možné použiť metriku LOC na zistenie stavu, v akom sa projekt nachádza. Avšak netreba výsledkom tejto metriky prisudzovať veľkú váhu. LOC treba brať ako taký prvotný a rýchly náhľad na to, či a ako pokročil priebeh prác v etape implementácie, najmä vďaka jednoduchosti jej merania. Ako smerodajnú metriku pre plnenie plánu v etape implementácie by som odporučil opäť použitie metriky funkčných bodov. Je z nej možné určiť, akým tempom sa doteraz vyvíjal produkt (počet implementovaných funkčných bodov za mesiac) a na základe toho presnejšie odhadnúť čas potrebný na implementáciu zvyšných funkčných bodov.

V každom produkte sa vyskytujú chyby, či už vyplývajú z nesprávnej interpretácie požiadaviek alebo nevhodnej implementácie. Aby bolo možné sledovať výskyt chýb v jednotlivých verziách aplikácie, resp. v niektorých jej moduloch, je potrebné zaviesť metriku na meranie chýb. Táto metrika pokrýva etapu testovania a nasadenia aplikácie. Porovnaním hustoty chýb na funkčný bod v jednotlivých verziách aplikácie je možné zistiť, akým tempom sa jednotlivé chyby odstraňujú, či vznikajú nové chyby a pod.

Jednotlivé využitie metrík je prehľadne zobrazené v Tab. 1.

Tab. 1. Prehľad využitia jednotlivých metrík.

Metrika	Využitie
Funkčné body	Odhad veľkosti produktu Kontrola vykonanej práce počas etapy implementácie Odhad času potrebného na dokončenie produktu
LOC	Rýchly náhľad pri kontrole vykonanej práce
Hustota chýb	Kontrola odstraňovania chýb

Záver

Použitie softvérových metrík je vhodným nástrojom na meranie v projekte. Avšak použitie jednej metriky nepostačuje na celkové pokrytie merania v rámci vývoja produktu. Vhodnou kombináciou týchto metrík je možné dosiahnuť želaný efekt merania. Na odhad veľkosti produktu je vhodné použiť metódu funkčných bodov, pretože vychádza priamo zo špecifikácie. Na sledovanie postupu prác počas implementácie je vhodné použiť ako východiskovú metriku opäť metódu funkčných bodov, ale na rýchly náhľad postačuje aj metrika LOC, najmä vďaka svojej jednoduchosti. Etapu testovania a nasadenia aplikácie pokrýva hustota chýb.

Použitá literatúra

1. Furey, S.: Why we should use function points. [online] *Software, IEEE* , vol.14, no.2, pp.28, 30, Mar/Apr 1997. Dostupné z <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=582971&isnumber=12658>
2. Jones, C.: Software metrics: good, bad and missing. [online] *Computer*, vol.27, no.9, pp.98-100, Sep 1994. Dostupné z http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=28084&tag=1

Annotation

How to combine software metrics?

Measurement in software project is the best way to identify project state during its realization. Software metrics are used for measurement in software projects. However, the real value of these metrics may be affected by the psychical effort of measurement on the people. This may leads to the state, when the development of product is oriented to reach better measurement result rather than to achieve a goal. In this paper I discuss suitability and usability of three frequently used software metrics: lines of code (LOC), function points and defect density. All of this metrics have their Pros and Cons. There is no metric that covers all phases of the software development. This is why it is necessary to combine them.