ZLEPŠOVANIE KVALITY KÓDU POMOCOU TECHNÍK REFAKTORINGU

Refaktoringom za kvalitou kódu.

Peter Greguš

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

Abstrakt. *Nie je jednoduché vytvoriť kvalitný kód. Dosiahnuť tento cieľ je možné pomocou viacerých spôsobov. Avšak kedy vieme, že už sme vytvorili kód v požadovanej kvalite? Odpoveď na túto otázku nám poskytujú metriky kódu, ktoré nám prezrádzajú aký kvalitný kód máme. Podľa hodnôt, ktoré tieto metriky nadobúdajú, vieme posúdiť kvalitu dvoch kódov. Ak ale chceme posúdiť kvalitu samostatného kódu, musíme poznať hodnoty metrík pre kvalitný kód, aby sme dokázali určiť, či je daný kód kvalitný alebo nie. Spôsob ako z nekvalitného kódu dosiahnuť kvalitný kód je využívanie techník refaktoringu. Pomocou tohto nástroja dokážeme meniť štruktúru kódu tým spôsobom, že vylepšujeme jeho štruktúru (kvalitu), pričom nemeníme jeho vonkajšie správanie. Aplikácia refaktorizačných pravidiel sa odzrkadlí na hodnotách metrík určujúcich kvalitu kódu, avšak nie je vhodné sa ich pevne držať, ale využívať ich ako indikátory prípadných potrebných zmien v kóde. Preto je vhodné využívať refaktoring pri indikácii zmeny nevhodnými hodnotami metrík a nie ako prostriedok na vylepšovanie týchto hodnôt, nakoľko sa nám môže kód znehodnotiť. Je preto potrebné vedieť odhadnúť správnu mieru využívania refaktoringu.*

Kľúčové slová: *kvalita kódu, metriky kódu, refaktoring*

Úvod

Ako dokážeme opísať čo je to kvalitný kód? Pri viacerých typoch produktov je kvalita meraná odlišne. Môže ju určovať viacero vlastností ako doba trvanlivosti, záručná doba, hodnotenia od externých inštitútov zaoberajúcich sa kvalitou výrobkov, spokojnosť zákazníkov a ďalších.

Kvalita produktu by mala byť odrazená v jeho výslednej cene tak, že ak vezmeme skupinu produktov z rovnakej triedy od iných výrobcov, tak drahšie produkty by mali byť kvalitnejšie. Samozrejme nemusí to byť vždy pravda, keďže v cene sú zahrnuté aj iné položky ako hodnota značky, hodnota použitého materiálu, hodnota práce atď., ale značnú mieru z ceny by mala reprezentovať kvalita.

Kvalitný kód

Pomenovať čo je kvalitný kód je pomerne náročné. Vo všeobecnosti pri rôznych produktoch je kvalita ocenená cenou, teda peniazmi, ktorými zaň zákazník zaplatí. Obdobne to môžeme tvrdiť aj o kóde, avšak tento pohľad odzrkadľuje iba správnosť kódu t.j. či spĺňa požiadavky, ktoré zákazník zadal.

Tento pohľad sa týka vonkajšieho hľadiska, z vnútorného hľadiska má kód však aj ďalšie vlastnosti, ktoré s kvalitou súvisia ako znovupoužiteľnosť, bezpečnosť, čitateľnosť, štruktúra a ďalšie. Celkovú kvalitu kódu teda vyjadríme ako súhrn kvalít jeho jednotlivých vlastností. Na vyjadrenie týchto vlastností existuje súbor metrík. Medzi základné metriky patria:

- LOC (Lines of code) – počet riadkov kódu
- NOM (Number of methods) – počet metód
- NOC (Number of classes) – počet tried
- NPM (Number of public methods) – počet verejných metód
- atď.

Tieto metriky ale ešte nevypovedajú o kvalite kódu, ale pomocou ich kombinácie dostaneme metriky, ktoré už dokážu vyjadriť kvalitatívne vlastnosti kódu:

- WMC (Weighted methods per class) – celková zložitosť triedy
- RFC (Responce for a class) – počet metód triedy a počet volaných metód
- LCOM (Lack of cohesion of methods) – vyjadruje podobnosť metód
- CBO (Coupling between objects) – previazanosť triedy s ostatnými triedami [2][4].
- atď.

Ktorý kód je kvalitnejší?

Na základe porovnávania metrík dokážeme určiť, ktorý z dvoch kódov je kvalitnejší. Dokážeme však určiť či konkrétny jeden kód je kvalitný alebo nekvalitný? Porovnať dve čísla a určiť ktoré z nich je „lepšie“ nie je náročné, ale určiť hranicu, kde končí nekvalitný kód a tam kde začína kód kvalitný je podstatne komplikovanejšie. Vyhodnotenie, kde sa

tieto hranice nachádzajú, závisí od typu projektu, použitého programovacieho jazyka a ďalších faktorov.

Čo je kvalitný kód?

Ako teda určiť hranice metrík pre kvalitný kód? Dokážeme ich určiť empiricky z už vytvorených kódov tak, že prizveme experta, ktorý určí, či konkrétny kód z celkovej množiny kódov je kvalitný alebo nekvalitný. Potom pre ne zmeriame metriky. Pri ich meraní sa hodnoty ustália v určitom intervale pre kvalitné aj nekvalitné kódy, t.j. určí sa „hranica" medzi kvalitou a nekvalitou. Pri ďalšom vývoji nových kódov sa snažíme dosiahnuť, aby hodnoty ich metrík spadali do „správnych" intervalov. Ak sa nám to podarí, môžeme prehlásiť o našom kóde, že je kvalitný. Ak nastane situácia, že hodnoty metrík nevykazujú hodnoty podľa našich predstáv, je potrebné urobiť zmeny v našom kóde a znovu zmerať metriky. Keď dosiahneme „lepšie" čísla, urobili sme krok ku kvalitnejšiemu kódu.

Ako vytvoriť kvalitný kód?

Medzi najpoužívanejšie spôsoby ako dosiahnuť kvalitný kód sú implementácia návrhových vzorov a testami riadený vývoj. Dodržiavaním týchto metód dokážeme zvyšovať kvalitu výsledného kódu v porovnaní s postupom, pri ktorom žiadny z uvedených prístupov nepoužívame. Vyplýva to z toho, že pri týchto metódach píšeme kód automaticky do šablón podľa logickej súdržnosti.

Ďalšou cestou ako je možné dosahovať kvalitný kód je dodržiavanie konvencií pre daný programovací jazyk ako aj rôzne interné dohody v rámci firmy. Pri dodržiavaní týchto princípov by sa mala zvýšiť čitateľnosť kódu.

Menej využívaným nástrojom ako dosiahnuť kvalitnejší kód je refaktoring. Manažéri môžu ľahko nadobudnúť dojem, že ak program funguje, nie je potrebné sa k nemu vracať a upravovať ho, čo je ale veľká chyba. Ak sa totiž kód v procese vývoja priebežne nereviduje, môžu sa v ňom začať objavovať duplicity, triedy, ktoré obsahujú funkcionalitu, ktorá nie je na vhodnej úrovni abstrakcie a rôzne ďalšie problémy tzv. pachy v kóde [1]. Má to za následok zhoršovanie štruktúry, čo sa nemusí prejaviť hneď pri uvedení produktu na trh, ale pri jeho ďalších modifikáciách, údržbe a tvorbe nových verzií. Ak v priebehu tvorby kódu podceníme refaktoring a nebudeme ho využívať, znižujeme si tak kvalitu vyvíjaného kódu, čo môže zvýšiť náklady na neskoršiu manipuláciu s ním.

Ako tieto postupy ovplyvňujú kvalitu kódu? Pri dodržiavaní implementácie vzorov, testami riadenom vývoji a dodržiavaní konvencií, sa metriky kódu automaticky približujú k správnym hodnotám. Pri refaktoringu sa tieto hodnoty upravujú postupne, nakoľko nevhodné hodnoty týchto metrík môžu mať za následok samotné uplatnenie refaktorizačných pravidiel, ktorých výsledkom je zmena hodnôt metrík, ktoré sa priblížia, prípadne dosiahnu hodnoty kvalitného kódu.

Refaktoring

Refaktoring môžeme definovať ako proces meniaci softvérový systém tak, že vylepšujeme vnútornú štruktúru kódu, pričom neovplyvňujeme jeho vonkajšie správanie [3]. Tento

princíp funguje na detekcii pachov v kóde a následnom postupe, ako daný pach odstránime. Tým, že meníme štruktúru kódu meníme aj čitateľnosť, z časti výpočtovú efektívnosť, iné charakteristiky kódu a metriky, ktoré určujú jeho kvalitu.

Proces refaktoringu

V procese refaktoringu meníme čitateľnosť kódu v tom smere, že „rozbíjame", zlučujeme a zoskupujeme kód na jednoduchšie časti, ktoré je potrebné logicky preusporiadať, meníme názvy premenných, metód atď. Dôsledkom týchto zmien je síce lepšia štruktúra kódu, ale zhoršujeme čitateľnosť tým, že rozdeľujeme väčší celok na menšie, ktoré nemusia byť „blízko seba" v kóde, ale sú rozdelené podľa pravidiel, logiky a úrovne abstrakcie na viaceré miesta v kóde. Takýto kód si bude pri potrebných zmenách vyžadovať dlhší čas na preštudovanie programátorom, avšak tie sa budú implementovať rýchlejšie. Musíme preto zvážiť, do akej miery budeme nástroje refaktoringu využívať.

Vplyv refaktoringu na metriky kódu

Pri metrikách je zmena štruktúry kódu najviac viditeľná. Pomocou nich vieme presne povedať, do akej miery nastali zmeny. Sledovať tieto metriky je potrebné, nakoľko nám vypovedajú o kvalite kódu, ale nie je dobré sa ich pevne držať a snažiť sa mať za každú cenu „dobré čísla", nakoľko môžu mať za následok nečitateľný a neefektívny kód.

Vo všeobecnosti má súbor metrík dobrú výpovednú hodnotu a ak nastane prípad, keď pri časti kódu – triede – metriky ukazujú zvýšené hodnoty (napr.: NOM, WMC, RFC), je zrejmé, že je potrebné túto triedu rozdeliť do viacerých, pretože je príliš zložitá. Vo väčšine prípadov je skutočne výhodné túto triedu rozdeliť na základe pohľadu z nižšej úrovne abstrakcie. Potom budú uvedené metriky ukazovať hodnoty vypovedajúce o kvalitnom kóde.

Nie vždy je ale vhodné „silou" presadzovať aplikáciu techník refaktoringu iba z dôvodu zlepšenia hodnôt metrík kódu. Metriky nám prezrádzajú, že sa v našom kóde vyskytujú nedostatky, ale je na manažmente a programátoroch, kedy je vhodné kód upravovať. Ako je vyššie spomenuté, vo väčšine prípadov skutočne je správne refaktorovať kód, ale môžu nastať aj také prípady, keď zmena v kóde zasiahne do výpočtovej efektivity samotného programu, alebo nastane iný neobvyklý prípad. V týchto špeciálnych prípadoch, napriek tomu, že sa neobjavujú často, je vhodné zvážiť, či nám zmeny v kóde, ktoré by sa odrazili do ideálnejších hodnôt metrík stoja za to, ak sa prejavia nežiadúcim spôsobom.

Vplyvy refaktoringu dobré vs. zlé

Využívanie refaktoringu v prvom rade napomáha udržovateľnosti a pružnosti kódu, nakoľko meníme jeho štruktúru tak, aby bol ľahšie uržovateľný a napomáha aj jednoduchším implementáciám zmien, prípadne novej funkcionality. To dosiahneme zavedením vzoru, ktorý usporiada kód do štruktúry, v ktorej je jednoduchšie zavádzanie ďalšej funkcionality. Do veľkej miery nám napomáha aj v procese znovupoužiteľnosti tak, že je možné vziať časť tohto kódu a použiť pri inom projekte, nakoľko vykonané zmeny v štruktúre umožňujú vziať ucelenú časť kódu, čo pri neudržiavanom kóde nie je také jednoduché. Miera znovupoužiteľnosti nám vypovedá o samotnej kvalite kódu.

Pri výpočtovej efektívnosti môže nastať posun oboma smermi pri používaní refaktoringu. Zmena k rýchlejšiemu spracovaniu je vítaná, naopak, ak by sa táto efektívnosť mala zhoršiť, je potrebné zvážiť použitie refaktoringu.

Naopak pri častom využívaní refaktorovania sa nám zhoršuje čitateľnosť. Pri jednoduchých zmenách ako premenovanie parametra, alebo rozbitie dlhej metódy, sa čitateľnosť zvýši, avšak pri zavádzaní väčších zmien ako preusporiadavanie kódu podľa komplikovaných vzorov a pod., sa nám čitateľnosť prudko zníži. Napríklad zavedením vzoru Template method, alebo aspektových vzrov nie je jednoduché porozumieť čo sa v kóde deje, pretože kód nemôžeme čítať sekvenčne pre jeho pochopenie, ale potrebujeme naštudovať viac metód, prípadne väčších úsekov kódu, ktoré nie sú umiestnené v jednej triede, alebo balíku. Je potrebné urobiť si celkový obraz o fungovaní kódu. Táto časť síce zaberie dlhší čas, ale doba implementácie novej funkcionality do kódu je podstatne kratšia v porovnaní s prípadom, keby sme mali kód napísaný procedurálnym štýlom.

Je preto na programátorovi, aby si dokázal zvážiť mieru využívania refaktoringu, nakoľko je potrebné poznať v daný moment plusy a mínusy, a na ich základe sa vedieť rozhodnúť v prospech celkovej kvality kódu.

Záver

Počas vývoja programu sa zdrojový kód vytvára, dopĺňa, rozširuje, mení, premiestňuje, prispôsobuje zmenám, atď. Preto, ak ho chceme udržať na vysokej úrovni kvality, musíme sa oň neustále starať a dozerať na jeho vývoj.

Na dozeranie nad vývojom nám slúžia metriky, ktoré hovoria o kvalite aktuálneho stavu vývoja kódu. Ak tieto metriky začnú indikovať že je potrebná zmena, je dôležité zvážiť využitie refaktoringu. Pomocou neho dokážeme kód udržiavať na požadovanej úrovni kvality.

Avšak najdôležitejšie je dokázať zhodnotiť situáciu a povedať, kedy refaktoring využiť a kedy nie. Aby sme nezískali iba kvalitný kód z hľadiska jeho metrík, ale aj kvalitný výsledný program, ktorý ponúkneme zákazníkovi.

Použitá literatúra

1. Long, T., Liu, L., Yu, Y., Wan, Z.: Assure High Quality Code using Refactoring and Obfuscation Techniques. *Fifth International Conference on Frontier of Computer Science and Technology* 2010.
2. Meirelles, P., Santos, C., Miranda, J., Kon, F., Terceiro, A., Chavez, C.: A study of the Realtionships between Source Code Metrics and Attractiveness in Free Software Projects. *Brazilian Symposium on Software Engineering* 2010.
3. Opdyke, W.F.: *Refactoring Object-Oriented Frameworks*. Doctoral thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1992.
4. Stroggylos, K., Spinellis, D.: Refactoring – Does it improves software quality? *Fifth International Workshop on Software Quality* 2007.

Annotation

Improving code quality by using refactoring techniques

It isn't simple to make quality code. We can reach this goal in several ways, but when we will know, that we created the code in required quality? Answer to this question is given by the code metrics, which tells us the level of quality of our code. Than we can tell which of two codes is qualitatively better, given their metrics values. But when we want to tell if single code has good quality or not we must know the values of the metrics for quality code. The way how to reach quality code from inferior code is to use the refactoring techniques. By using this tool we can change code structure in a way, that we upgrading its internal structure (quality), but we don't change its external behavior. Applying the refactoring rules changes the values of the metrics, which set the code quality and we shouldn't bind ourselves to them, but use them as indicators, that we should change something in our code. Because of it we should use refactoring when the code metrics indicate bad values and not as engine to improve those values, because we can degrade our code in this way. So it's necessary to know when to use refactoring in the correct way.