ZABEZPEČOVANIE KVALITY SOFTVÉRU A REFAKTORING

Pri kvalitnom prístupe môžeme očakávať kvalitný výsledok.

Maroš Jendrej

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
maros136[zavináč]gmail[.]com

Abstrakt. *V eseji sa zaoberám kvalitou softvérových systémov. Poukazujem na prostriedky pomocou, ktorých možno zabezpečovať kvalitu softvéru. Medzi hlavné prostriedky na zabezpečovanie kvality patrí predovšetkým testovanie. Poukazujem tu na dôležitosť fáz ešte pred začiatkom testovania. Zabezpečenie kvality nie je len samotné otestovanie softvérového projektu, ale aj príprava kritérií a požiadaviek, ktoré majú byť podľa stanovenej úrovne splnené. Jeden z ďalších prostriedkov, ktorým dosiahneme lepšiu kvalitu v rozbehnutom softvérovom projekt je refaktoring. Pomocou neho sa mení architektúra softvéru bez toho, aby sa zmenilo jeho správanie a pritom sa zvýši jeho kvalita. Aplikovanie refaktoringu v projekte, však nezaručuje zvýšenie kvality. Existujú rôzne ukazovatele, pomocou ktorých zisťujeme či je refaktoring vhodný a do akej miery ho je možné uplatniť.*

Kľúčové slová: *kvalita softvéru, zabezpečovanie kvality, testovanie, refaktoring*

Úvod

V každom softvérovom projekte bez ohľadu na jeho veľkosť musíme zabezpečovať jeho kvalitu. Kvalita softvérových projektov je štandardizovaná vo viacerých medzinárodných normách. Spomeniem niekoľko týchto noriem vzhľadom nato, že jasne a zreteľne pomenúvajú hlavnú časť problematiky.

Podľa normy ISO 8402 je kvalita súhrn vlastností a charakteristík výrobku, procesu alebo služby, ktoré preukazujú jeho schopnosť splniť určené alebo odvodené potreby [4].

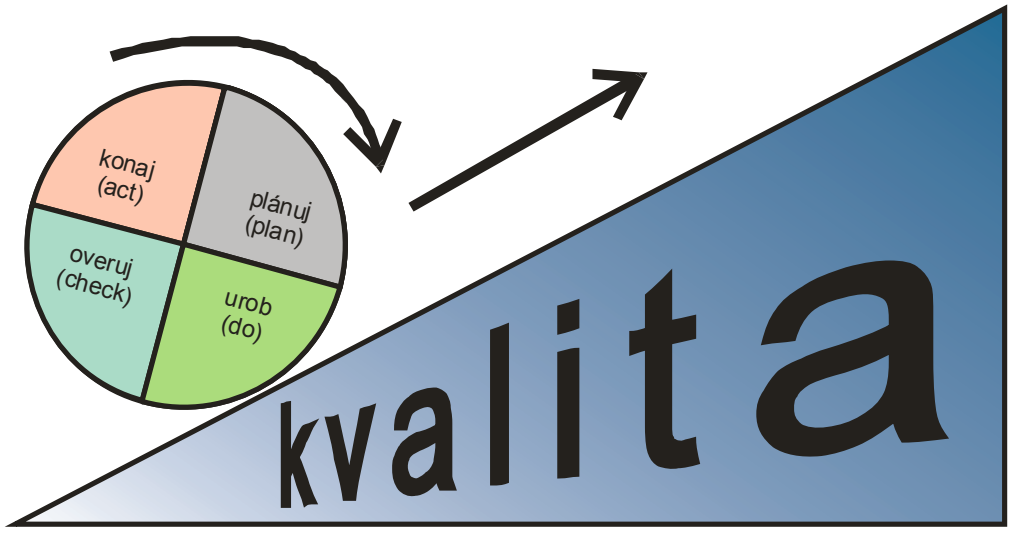
Podľa normy ISO 9000 je kvalita stupeň, ako sú splnené potreby a očakávania zákazníka [5]. Iné definície hovoria o tom, že kvalita je miera splnenia troch faktorov, ktorými sú požiadavky, čas a zdroje [2]. Nemožno ju preto merať, lebo rovnakú kvalitu môže mať aj výrobok, ktorý ma iný pomer požiadaviek, času a zdrojov. Z týchto definícií môžeme dedukovať, že kvalita sa zaoberá abstrahovaním určitých faktorov, podľa ktorých zákazník určuje svoju spokojnosť.

Prečo teda treba dbať na kvalitu v softvérovom projekte? Tak ako aj produkty v iných oblastiach aj tento softvérový produkt musí byť podľa požiadaviek dostatočne spoľahlivý, stabilný, bezpečný, flexibilný, výkonný či robustný. Na vývoj takéhoto softvérového produktu máme v softvérovom projekte len vymedzené časové obdobie a určité zdroje, ktoré musíme využívať tak aby sme dosiahli, čo najväčšiu spokojnosť zákazníka [3]. Zákazník náš produkt oceňuje, ak splnil zadané požiadavky za stanovených podmienok. Teda čím viac je produkt kvalitnejší, tým sú uspokojené potreby zákazníka.

Ako teda v softvérovom projekte uspokojiť zákazníka a vytvoriť kvalitný produkt? Tieto otázky riešime v manažmente kvality softvéru.

Procesy manažmentu kvality

V softvérových projektoch existujú procesy manažmentu kvality, ktoré sa starajú o to aby sme vytvárali kvalitný softvér už od jeho začiatku až do konca projektu. Sú to procesy plánovania, zabezpečovania a riadenia kvality [2]. Je vhodné poukázať na to, že tieto procesy pripomínajú Demingov cyklus, v ktorom neustále prebieha plánovanie (plan), vykonávanie (do), kontrolovanie (check) a konanie zmien (act). Myslím si, že ak v projekte chceme neustále a systematicky zvyšovať jeho kvalitu musíme dodržiavať tento postup (pozri Obr. 1).



Obr. 1. Demingov cyklus a kvalita.

Je to predovšetkým preto, že každú činnosť je veľmi vhodné naplánovať v širšom kontexte. Následne ju vykonávame, kontrolujeme jej výsledok a nakoniec zhodnocujeme jej dopady a následky. Proces zhodnocovania ovplyvňuje aj ostatné procesy, lebo z jeho analýzy často krát dokážme identifikovať zmeny potrebné na zlepšenie ostatných procesov.

V procese plánovania kvality definujeme proces zabezpečovania kvality a požiadavky na softvérový produkt. Definujeme postupy, pravidlá a dokumenty, ktoré sa využívajú v ďalších procesoch.

Proces zabezpečovania kvality softvéru je súbor aktivít, ktorých cieľom je tvorba kvalitného softvéru. V tomto procese sa priamo vykonávajú úkony zabezpečujúce kvalitu počas celého životného cyklu softvérového produktu v súlade zašpecifikovanými požiadavkami.

V procese riadenia kvality sledujeme projekt, dodržiavanie noriem a definovaných kritérií. Riadenie vykonávame hlavne na základe analýz a meraní v projekte, podľa ich výsledkov vykonávame potrebné kroky pre zlepšovanie ostatných procesov.

Zabezpečovanie kvality

Bez rozdielu veľkosti softvérového projektu je nutné zabezpečovať jeho kvalitu. Vo väčších projektoch je na to vyhradené väčšie množstvo prostriedkov ako v menších projektoch. V menších projektoch sa zase tieto prostriedky dokážu využívať efektívnejšie, napríklad pomocou agilných metód vývoja.

Jedným z jeho cieľov pri zabezpečovaní kvality je prevencia vzniku chýb tým, že sa neustále zdokonaľujú procesy tvorby softvéru. Myslím si, že je vhodné už v začiatočných etapách projektu dostatočne dbať na kvalitný vývoj softvéru. Predovšetkým vytváraním špecifikácie a modelov podľa jasne definovaných procesov a pravidiel. Špecifikáciu a modely treba neskôr overovať, či stále spĺňajú definované pravidlá.

V neskorších fázach projektu sa na zabezpečovanie kvality využíva verifikácia a validácia. Medzi hlavné prostriedky, ktoré sa na to využívajú je testovanie a refaktoring.

Čo a prečo testovať?

Testovanie je jeden zo základných spôsobov na zabezpečovanie kvality softvéru. Avšak pred začatím každého testovania je nutné si jasne vymedziť požiadavky a kritéria, ktoré treba na splnenie testov [3].

Pred začatím každého testovania musí byť vypracovaný podrobný plán, v ktorom je jasne definovaný objekt testovania a jeho výsledok. Takýto plán sa musí zameriavať predovšetkým na tieto faktory projektu:

- Veľkosť projektu
- Metódy testovania
- Použiteľnosť
- Prostredie
- Manažment rizík [1].

Môžem tvrdiť, že bez takéhoto plánu nie je testovanie dostatočne účinné. Je to hlavne preto, že softvér je veľmi zložitý a bez detailnejšej analýzy sa môže stať, že prehliadneme

kritické kombinácie, ktoré spôsobujú chybu. Tento plán určite nemusí obsahovať kompletný postup ako sa má projekt otestovať. Veď kompletné pokrytie veľkých projektoch ani nie je možne. Tento plán má byť optimálny zo zameraním na problematické oblasti a práve tieto faktory nám k tomuto napomáhajú.

Výsledkom testovanie je odhalenie chýb, ktoré sa nachádzajú v softvéri. Okrem tohto primárneho cieľa hľadania a odstraňovania chýb je potrebné vylepšenie procesu a zabezpečenie prevencie pred možným zopakovaným chýb. Napríklad ak chyba vzniká v dôsledku zlej flexibility kódu, tak treba tento faktor minimalizovať a zamedziť tak vzniku chyby. V takomto prípade je vhodné využiť refaktoring.

Existuje ešte jedna metóda na zabezpečovanie kvality softvéru je ňou prehliadka kódu (Code review). Tuto metódu považujem ako prechod medzi manuálnym testovaním (biela skrinka) a refaktoringom. Lebo človek, ktorý prezerá kód vlastne nachádza chyby ako pri testovaní. Zároveň sa zamýšľa nad prípadným vylepšením architektúry, tak aby sa v budúcnosti predišlo týmto chybám, podobne ako v refaktoringu.

Pomôže nám refaktoring?

Refaktoring je jeden z prostriedkov ako zlepšovať kvalitu v rozbehnutých softvérových projektov. Je to transformácia kódu do jeho efektívnejšej podoby, s ktorou sa pracuje rýchlejšie a lepšie. Nemusí to znamenať zlepšenie výkonnosti, ale predovšetkým jeho modifikovateľnosti a zníženie jeho zložitosti.

Jeho výhodou je, že ho môžeme vykonávať po malých krokoch, aby sa predišlo znefunkčneniu celého projektu. Zmena malých častí kódu sa dá rýchlo vrátiť späť pri prípadnom znefunkčnení kódu.

Prečo je potrebný refaktoring? Pomocou neho dokážeme zlepšovať predchádzajúci návrh softvéru, čitateľnosť kódu a umožňuje nám ľahšie odhaľovanie chýb pri ďalších zmenách. Všetky tieto zmeny priaznivo ovplyvňujú ďalšiu prácu zo zdrojovým kódom, takže sa tým zvyšuje aj celková kvalita projektu.

V zdrojovom kóde sa odhaľuje na základe „pachov“ sú to miesta, ktoré by benefitovali pri využití refaktoringu. Hľadanie týchto pachov je vo väčšine prípadov na ľudskej intuícií, avšak sú nám aj známe rôzne softvérové metriky, ktoré nám toto hľadanie uľahčujú. Okrem týchto metrík na odhaľovanie pachov v kóde, existujú aj sofistikovanejšie, ktoré dokážu zisťovať kvalitu kódu. Ukážme si teda niektoré metriky na využívané na zisťovanie kvality kódu:

- WMC (Weighted Methods per Class) – váha metód triedy
- DIT (Depth of Inheritence Tree) – hĺbka v hierarchii dedičnosti
- NOC (Number Of immediate Children subclasses) – počet potomkov
- CBO (Coupling Between Objects) – väzby medzi triedami
- RFC (Response for a class) – počet odpovedí triedy
- LCOM (Lack of Cohesion of Methods) – nedostatok súdržnosti metód [6].

Vidíme, že metriky podľa rôznych kritérií napomáhajú k zisťovaniu celkovej kvality softvéru, avšak takýto spôsob ohodnocovania kvality je jednostranný a do úvahy neberie aj možnosti zlepšenia sa iných softvérových procesov (riadenia, plánovania, atď.). Tieto metriky skôr slúžia na odhaľovanie kvality malých kúskov kódu ako na ohodnotenie

celkovej kvality. Lebo nie sú schopné analyzovať hlbšie vzťahy v rámci celého projektu, hovoria len o tom či sa časť kódu podobá na určitú zlú vzorku. Avšak si myslím, že je dobre ich využiť pri rozhodovaní, či začať refaktoring, ale len do určitej miery a nepovažovať ich za smerodajné.

Z tohto nám vyplýva, že aj keď nám metriky hovoria aby sme refaktorovali nemusí to byť zárukou zlepšenia kvality softvéru. Všeobecne platí, že ho je zbytočné vykonávať v projektoch, ktoré obsahujú veľké množstvo chýb a sú tiež príliš chaotické. V takýchto projektoch je lepšie začať odznova a vyvarovať sa chybám už hneď od začiatočného návrhu. Problémom však môže byť ak projekt je tak rozsiahly, že nemožno začať odznova a je nutné ho udržiavať vo funkčnom stave. V takomto prípade sme nútení využiť refaktoring na zlepšenie kvality softvéru. Treba preto dôkladne premyslieť v akom rozsahu zasiahne projekt a koľko úsilia naň bude potrebné vynaložiť. Práve v takýchto prípadoch považujem využitie metrík na zisťovanie kvality kódu ako vhodný prostriedok pre doplnkový prieskum, ktoré časti softvéru sú najkritickejšie.

Záver

V eseji som sa zaoberal kvalitou softvéru a poukázal som na prostriedky zabezpečovania kvality v softvérových projektoch. Jeden z takýchto prostriedkov je testovanie, pri jeho aplikácií je dôležité ešte pred začatím vytvárania testov stanoviť si jasné kritéria a požiadavky. Následne z nich vytvoriť podrobný plán testovania, ktorý zahŕňa aj kritické faktory v projekte. Ukázal som tiež refaktoring ako jeden z prostriedkov zabezpečovania kvality pre rozbehnuté projekty a diskutoval som o využití softvérových metrík na zisťovanie kvality kódu. Zistil som, že jeho použitie nie je vždy vhodné a nemusí znamenať zlepšenie kvality. Treba si preto dať pozor, kedy a ako ho v rozbehnutom projekte využijeme. Lebo zbytočné vynakladanie zdrojov a úsilia kvalitu projektu len zníži.

Použitá literatúra

1. Alsultanny, Y.A., Wohaishi. A.M.: *Requirements of Software Quality Assurance Model*, In Proceedings of the 2009 Second International Conference on Environmental and Computer Science, IEEE Computer Society, Washington DC, 19-23, 2009.
2. Bieliková, M.: *Softvérové inžinierstvo: Princípy a manažment,* FEI STU Bratislava, ISBN 80-227-1322-8, 2000.
3. Feldman, S.: *Quality Assurance: Much More than Testing,* Queue 3, No. 1 (2005) 26-29.
4. ISO 8402 Quality management and quality assurance, International Standard Organization.
5. ISO 9000 Quality Management Systems – Guidelines for Performance Improvements, International Standard Organization.
6. Stroggylos, K. Spinellis, D.: *Refactoring – Does it improve software quality*, WoSQ '07 Proceedings of the 5th International Workshop on Software Quality, Conference ICSE International Conference on Software Engineering, 10-16, 2007.

Annotation

Software quality assurance and refactoring

In an essay dealing with the quality software of systems. I point to resource by which software can ensure software quality. The main resource for quality assurance is testing. I show here to the importance of phases before the start of testing. Quality assurance is not merely testing the software project, but also to prepare the criteria and requirements to be met by specified levels. One of the other resources for the better the quality of the software project is refactoring. Refactoring is changing architecture of the software without having to change his behavior, while increasing its quality. Applying refactoring in the project, but does not improve quality. There are various indicators by which detects whether the refactoring is appropriate and as possible is applicable.