

ZACHRÁNI TESTOVANIE KVALITU?

*Čím viac sa tester raduje, tým viac sa programátor
hnevá.*

Róbert Korduliak

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

Abstrakt. *Kvalita v dnešnej dobe vyspelej konkurencie a uvedomelých zákazníkov predstavuje často opakované slovo na mnohých firemných poradách. Rovnaká situácia je aj v softvérovej oblasti. Firmy sa snažia vyvíjať kvalitný softvér aby uspokojili požiadavky zákazníka, získali si dobré meno, primeraný zisk a aby minimalizovali náklady na riešenie problémov súvisiacich s nekvalitným softvérom. Samotný pojem kvalitný softvér je však ťažké definovať a samozrejme existuje veľa spôsobov, ako ho vytvoriť. Myslím si, že je chybou ak sa testovanie kladie ako posledná inštancia, ktorá by mala zaručiť vytvorenie kvalitného produktu. V tejto eseji sa preto zameriam na vzťah kvality a testovania. Pokúsim sa spresniť pojem kvalitný softvér a bližšie špecifikovať úlohu testovania v procese vývoja softvéru. Táto špecifikácia by mala zabrániť nereálnym očakávaniam a nevhodnému nasadeniu testovania do vývoja produktu.*

Kľúčové slová: *kvalita, testy, testovanie, plánovanie, riadenie kvality*

Úvod

Kvalita je pojem, ktorý rezonuje vo veľmi širokom spektre oblastí dnešného života. Či už sa jedná o priemysel, techniku, zdravotníctvo, školstvo, informačný sektor. Kvalita sa dnes často ohýba podľa ekonomických ukazovateľov. Firmy v mnohých oblastiach v snahe maximalizovať zisk upúšťajú od kvality výrobkov či služieb. Ako je to v informačnom sektore? Snažia sa firmy o vytváranie kvalitného softvéru? Myslím si, že firmy v informačnom sektore sa snažia o dosiahnutie aspoň určitej úrovne kvality. Niektoré sa

snažia podriaadiť kvalitu nákladom na výrobu softvéru a snažia sa kvalitu dosiahnuť pomocou testovania. Podľa môjho názoru je tento prístup nesprávny a v tejto eseji sa pokúšam vysvetliť prečo. V prvom rade je dôležité charakterizovať pojmy testovanie a kvalita. Mojm cieľom je oddeliť význam testovania od významu kvality a chcem ukázať, že nielenže nie je možné samotným testovaním zabezpečiť kvalitu softvéru, ale navyše je takáto snaha vo väčšine prípadov i ekonomicky nerentabilná.

Čo je to kvalita?

Samotné slovo kvalita predstavuje široký pojem dá sa pod ňou predstaviť mnoho aspektov, napr. spoľahlivosť, merateľnosť, stabilita, príjemné používanie, ... [2]. Na slovné spojenie kvalitný produkt, resp. kvalitný softvér existuje taktiež mnoho pohľadov. Môžeme ho chápať ako taký, ktorý je správny, spoľahlivý, jednoduchý, robustný, znovu použiteľný, presný, dobre zdokumentovaný... Kvalita vyjadruje vo všeobecnosti mieru splnenia požiadaviek [1]. Ak by bolo našou požiadavkou dostať nepresný a nespoľahlivý softvér, práve takýto bude kvalitný. Z toho vyplýva, že ak chceme dosiahnuť kvalitu produktu, musíme čo najlepšie splniť požiadavky zákazníkov (aj nevyslovené a nezdokumentované požiadavky – očakávania).

Čo je to testovanie?

V jednoduchosti by sa dalo povedať, že testovanie je proces, pri ktorom tester kontroluje softvér s cieľom overiť, či spĺňa deklarované požiadavky. Autor knihy Testování softwaru Ron Patton uvádza nasledovnú definíciu úlohy testera: „Cieľom softvérového testera je vyhľadávať chyby, vyhľadať ich čo najskôr a zaistiť ich opravu“ [3]. Testovanie je teda jednoducho povedané hľadanie chýb. Tieto chyby je možné hľadať rôznymi spôsobmi. Testovanie môže byť statické, dynamické, testovanie čiernej, bielej a šedej skrinky, testovanie konfigurácie, kompatibility, lokalizácie, použiteľnosti, dokumentácie, alfa a beta testovanie, regresné testovanie[5]. Jednotlivé metódy je možné rôzne kombinovať podľa toho, aké aspekty testovania sa sledujú a aká časť vývoja sa testuje.

Prečo je potrebné vytvárať kvalitný softvér?

Predtým než rozoberieme samotnú myšlienku eseje, vzťah testovania a kvality, zamyslime sa, prečo vlastne potrebujeme kvalitný softvér. Mnoho používateľov stále v značnej miere akceptuje nekvalitný softvér. Ak nám zamrzne operačný systém, nejdeme ho hneď reklamovať u výrobcu. Ak nám aplikácia zničí dáta, budeme ich nabudúce lepšie zálohovať. Ak nejde nejakú webovú stránku otvoriť v istom prehliadači, vyskúšame iný. Prečo je teda tak dôležité snažiť sa dosahovať kvalitu? Objasnenie prinesie pohľad z iného uhla. Zoberme si softvér, ktorý sa má len vyvinúť a zákazník si má vybrať firmu, ktorá ho vytvorí. Zrejme jedným z kritérií bude spokojnosť zákazníkov, pre ktorých už firma pracovala. Softvér navyše už nie je výsadou len niekoľkých vybraných spoločností, ale masovo preniká do všetkých oblastí. Na trhu dnes pôsobí množstvo softvérových spoločností, ktoré sa predbiehajú v tom, aby ponúkli kvalitnejší produkt ako konkurencia.

Môžeme si ďalej pripomenúť mnohé prípady z histórie, keď nekvalitný softvér zapríčinil veľké problémy.

- Problém s prechodom na rok 2000(Y2K)¹
- Strata pristávacieho modulu na Marse²
- Smrť pacienta, ktorý dostal kvôli chybe softvéru smrteľnú dávku žiarenia³

Potrebu vytvárať kvalitný softvér teda zabezpečuje hlavne konkurencia medzi firmami a snaha vyhnúť sa uvedeným nešťastiam.

Zachráni testovanie kvalitu?

V predchádzajúcich odsekoch sme si ujasnili pojmy kvalita a testovanie a uviedli dôvody, prečo vytvárať kvalitný softvér. Až teraz sa teda môžeme zamyslieť nad hlavnou myšlienkou tejto eseje: „Zachráni testovanie kvalitu?“

Podľa môjho názoru testovanie nezachraňuje a ani nemôže zachraňovať kvalitu softvéru. Dalo by sa namietat, veď predsa keď testerí nájdu chyby, tieto sa opravujú a zákazník teda nedostane chybný produkt. Táto poznámka je naozaj pravdivá. Vráťme sa však opäť k pojmu testovanie a k úlohe testera. Pre lepšie pochopenie mojej myšlienky uvediem ešte raz definíciu od Rona Pattona: „Cieľom softvérového testera je vyhľadávať chyby, vyhľadať ich čo najskôr a zaistiť ich opravu“[3]. Cieľom testera (môžeme to zovšeobecniť na cieľ testovania) je hľadať chyby. Výstupom testovania sú teda nájdené chyby, nie kvalitný softvér. Toto je veľmi podstatný rozdiel, ktorý si mnohí ľudia (žiaľ často aj vo vedení firiem) neuvedomujú. Posledná časť definície („... zaistiť ich opravu“) predstavuje nahlásenie nájdených chýb kompetentným osobám a snaha testera presvedčiť tieto osoby o dôležitosti opravy daných chýb. Nie vždy sa totiž nájdené chyby opravujú (napr. je to príliš drahé, alebo by to príliš narušilo iné časti). Testerí navyše štandardne nezasahujú priamo do procesu vývoja softvéru a nemajú ani kompetencie ovplyvniť spôsob vývoja. Podľa spôsobu nasadenia testovania vo vývoji proste testerí hľadajú chyby (či už v špecifikácii, implementácii, integračnom alebo akceptačnom testovaní) a s väčším alebo menším úsilím ich nachádzajú. Takto postavená úvaha dáva logické dôvody, prečo testovanie nezachráni kvalitu softvéru. Pre lepšie pochopenie môžeme situáciu pripodobniť k chorému pacientovi u lekára. Lekár pravdepodobne nevylieči pacienta iba tým, že mu odmeria všetky symptómy.

Môžeme si ešte raz postaviť otázku: „Keď teda testerí nájdu všetky chyby, tieto sa odstránia, bude už softvér kvalitný“? Odpoveď na takto položenú otázku nie je jednoduchá. Môžu nájsť testerí všetky chyby? A aké chyby vôbec máme na mysli? Ron Patton opäť uvádza výbornú definíciu chyby (softvérovej) [3]:

- Softvér nerobí niečo, čo by podľa špecifikácie robiť mal
- Softvér robí niečo, čo by podľa špecifikácie robiť nemal
- Softvér robí niečo, čo v špecifikácii nie je

¹ <http://www.infostat.sk/y2k/>

² <http://marsprogram.jpl.nasa.gov/msp98/lander/>

³ http://courses.cs.vt.edu/~cs3604/lib/Therac_25/Therac_1.html

4 Róbert Korduliak

- Softvér nerobí nič, čo síce v špecifikácii nie je, ale malo by byť
- Softvér je ťažko zrozumiteľný, ťažko sa s ním pracuje, je pomalý, alebo podľa názoru testera nebude zákazníkovi vyhovovať

Ak by sme uvažovali iba chyby v programovom kóde, pri naozaj dôkladnom otestovaní by bolo možné nájsť a opraviť všetky chyby. I to by ale nemuselo priniesť kvalitný softvér, pretože by bol napr. pomalý a zákazník by bol s ním nespokojný. Z uvedenej definície chyby je ešte zrejmejšie, že je asi nereálne očakávať od testera, že nájde všetky možné chyby. Navyše, zákazník môže ako chybu vnímať nejakú vlastnosť softvéru, ktorá nebola predmetom špecifikácii, nebola nikde spísaná. Vhodný príklad je napr. odozva systému – ak sa pri analýze nestanovilo merateľné kritérium pre odozvu systému, darmo sa môže firma obhajovať, že otestovala celý produkt. Zákazník bude nespokojný so softvérom, a ťažko ho v takom prípade môže hodnotiť ako kvalitný.

Ako zachrániť kvalitu softvéru

V predchádzajúcom odseku sme si objasnili dôvody, prečo testovanie nemôže zachrániť kvalitu. Chýba tu však ešte opačný pohľad. Ako môžeme zachrániť, zabezpečiť, dosiahnuť kvalitu? Dá sa to vôbec?

Podľa môjho názoru je veľmi ťažké, možno až nemožné zaručiť 100% kvalitu softvéru. Dnešný softvér je zväčša až príliš veľký, resp. zložitý nato, aby sa v ňom dali skontrolovať všetky aspekty kvality do posledného detailu. Inapriek tomu ale mnohé firmy pochopili dôležitosť vývoja kvalitného softvéru a snažia sa preto o zvyšovanie kvality svojich produktov. Aj keď možno nezaručia 100% kvalitu, dosiahnu ale takú úroveň, že zákazník bude s produktom spokojný a označí ho za kvalitný.

Ak chceme oddeliť testovanie od kvality, musíme poznať spôsoby, ako kvalitu dosahovať. Ak budeme poznať tento pohľad, ešte jednoduchšie pochopíme prečo testovanie nezachráni kvalitu, pretože budeme vedieť ako ju dosiahnuť.

Autorka knihy Řízení projektu v IT, Kathy Schwalbe [4] dáva do popredia proces Riadenia kvality. Úlohou riadenia kvality v projekte je: „Zaistiť, že projekt uspokojí potreby, kvôli ktorým bolo začaté jeho riešenie“[4]. Riadenie kvality pozostáva z nasledujúcich procesov:

- Plánovanie kvality
- Zaisťovanie kvality
- Kontrola kvality

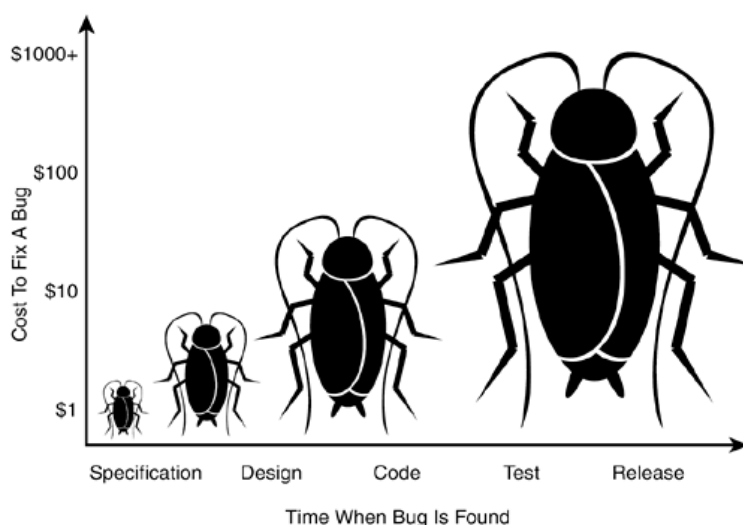
Plánovanie kvality je podľa Schwalbeovej vymedzenie noriem a štandardov kvality, ktoré sú pre daný projekt relevantné. Dôležité je tieto normy a štandardy zaradiť aj do návrhu samotného projektu. Medzi tieto normy môže patriť napríklad aj to, že v ňom budeme uvažovať o budúcom rozšírení systému alebo stanovíme presné hodnoty odozvy systému.

Zaisťovanie kvality zahŕňa: „Všetky aktivity, ktoré súvisia s uspokojením príslušných noriem a štandardov kvality projektu“[4]. Obsahuje teda postupy a metódy, ktorých dodržiavanie alebo kontrolovanie by malo zaistiť plánovanú kvalitu. Patria tu napr. porovnávacie testy (napr. aký rýchly je softvér konkurenčnej firmy), audity kvality. Kontrola kvality slúži na hodnotenie, v akej miere proces zaisťovanie kvality napĺňa plán

kvality. Priamo sa tu sledujú výsledky projektu, ktoré môžeme porovnávať s plánom kvality. Existuje viacero metód pre kontrolu kvality, spomeniem napr. Paretovu analýzu (20% príčin produkuje 80% chýb), štatistické vzorky (výber pre skúmanie kvality). Súčasťou kontroly kvality môžu byť tiež preventívne alebo nápravné opatrenia, ktoré môžu včas zvrátiť nekvalitný vývoj softvéru.

Koľko stojí kvalita?

Teraz sa môžeme zamyslieť nad ekonomickým faktorom výroby kvalitného softvéru. Neoplatí sa firmám z finančného hľadiska nestarať sa priebežne o kvalitu vývoja, ale investovať radšej prostriedky do konečného otestovania a odstránenia chýb? Do priebežného riadenia kvality totiž možno zaradiť prácu manažérov, vedúcich ktorý majú na starosti plánovanie a zaisťovanie kvality, pracovníkov ktorí majú na zodpovednosť kontrolu dodržiavania kvality, zavedenie rôznych noriem (napr. zavedenie ISO 9000 iste nie je lacná záležitosť). Oproti tomuto je zaplatenie aj väčšieho množstva hodín práce testera oveľa lacnejšie. Iste sme si však vedomí (aj z predchádzajúceho kontextu), že tester svojím testovaním nemôže plne nahradiť proces riadenia kvality. Túto myšlienku môžem podporiť teóriou Američana Philipa Crosbyho, autora mnohých kníh zaoberajúcich sa kvalitou. Základom jeho teórie je jeho výrok: „Kvalita je zadarmo“. Ako môže byť kvalita zadarmo? Ron Patton [3] sa pokúša jeho výrok vysvetliť porovnaním nákladov na predchádzanie chýb a nákladom na opravu chýb.



Obr. 1. Cena chyby v životnom vývoji softvéru [3]

Obrázok 1 vyjadruje rast cien opravy chýb v závislosti od času ich nájdenia. Čím skôr sa chyba nájde, teda napr. ešte pri špecifikácii, alebo dizajne produktu, tým menšie náklady sú na jej opravenie. Chyba, ktorá sa dostane až ku zákazníkovi, predstavuje často dodatočnú špecifikáciu, zmeny v kóde, nové otestovanie, náklady na vydanie novej verzie, a teda v konečnom dôsledku mnohonásobné predraženie opravy. Ron Patton delí náklady do dvoch kategórií – *náklady pri zhode* a *náklady pri nezhode*. Náklady na plánovanie,

jednorazové otestovanie, či produkt spĺňa to čo má, patria do prvej kategórie. Pokiaľ sa ešte pri vývoji produktu a jeho testovaní nájdú chyby, náklady na ich opravu sa radia k nákladom pri nehode. Dosať spomenuté náklady však prebiehajú v čase vývoja, a preto ich označuje ako *interné náklady*. Tieto spadajú do ľavej časti obrázka č.1. Ak sa však chyby dostanú až k zákazníkovi, ide o *externé náklady* a tieto patria do pravej časti obrázka. Phil Crosby hovorí, že súčet interných nákladov (pri zhode aj nezhode) je vždy menší, ako externé náklady. Pokiaľ sa teda chyby opravujú už v skorých začiatkoch, prípadne ak žiadne nie sú, je to lacnejšie ako oprava chýb zistených zákazníkom. Z toho teda vyplýva, že chyby ktoré by sa opravili na konci vývoja pomocou testovania by stáli firmu viac prostriedkov, ako zavedenie riadenia kvality a prevencia pred chybami.

Pre lepšie pochopenie uvediem ešte nasledovný príklad. Ak firma pracuje na produkte a má ho odovzdať v stanovenom termíne. V pôvodnom pláne síce bol vyčlenený dostatočný čas na testovanie a kontrolu štandardov, ale implementácia sa pretiahla a preto sa poľavilo v kontrole. Firma síce odovzdá produkt zákazníkovi, a ušetrí tak krátkodobé náklady, ktoré by vznikli pri posunutí času odovzdania. Ak však zákazník objaví nejaký závažný nedostatok, firma ho musí odstrániť, čo ju v konečnom dôsledku stojí niekoľkonásobne viac. Keby teda vydala kvalitný produkt, vyšlo by ju to lacnejšie ako keď vydala produkt nekvalitný. Musela teda zbytočne vynaložiť väčšie prostriedky na zabezpečenie kvality. Myslím si že uvedené myšlienky sú dostatočným dôvodom nato, aby sme sa nepokúšali pri vývoji softvéru nahradiť celý proces riadenia kvality iba testovaním softvéru.

Testovanie v procese vývoja softvéru

Posledný pohľad tejto eseje je umiestnenie testovania do vývoja softvéru. Pokiaľ nám bude jasné využívanie testovania pri vývoji softvéru, nebudeme sa snažiť nahradiť ním riadenie kvality. Testovanie by podľa môjho názoru malo prebiehať celým vývojom softvéru. Testovať možno špecifikáciu a analýzu, jednotlivé komponenty softvéru a ich súčinnosť (integračné testovanie), testovanie kompletného softvéru (testovanie systému) a testovanie prijatia zákazníkom (akceptačné testovanie). Uvedené princípy znázorňujú širokú pôsobnosť testovania. Takéto testovanie už v priebehu vývoja môže odhaliť chyby, ktorých náprava v neskorších etapách vývoja by bola veľmi nákladná. Nato, aby bolo možné testovanie efektívne využiť, je dôležité ho vhodne naplánovať už pri začiatku vývoja softvéru. Práve to je okamih, kde je potrebné rozlíšiť úlohu testovania a úlohu riadenia kvality. Testovanie (i v priebehu vývoja) sa zaoberá hľadaním chýb a ich opravou. Riadenie kvality sa ale zaoberá názormi a požiadavkami zákazníka, aby bol s daným softvérom po jeho dokončení spokojný. Z uvedeného pohľadu vyplýva podľa môjho názoru fakt, že testovanie sa zaoberá naozaj iba softvérom, jeho chybami a nie spokojnosťou zákazníka s konečným produktom.

Záver

Vytváranie kvalitných produktov je dnes už zrejme cieľom každého vývojového tímu, firmy či organizácie. Nie vždy sú však jasné postupy, ako kvalitu dosahovať a zabezpečovať. Je dôležité uvedomiť si rozdiel medzi riadením kvality a jeho procesmi

a samotným testovaním. Ak firma vyvinie softvér a jediný spôsob, ako chce zabezpečiť jeho kvalitu je, že ho otestuje pred odovzdaním zákazníkovi, zrejme vynaloží značné finančné prostriedky a výsledok i napriek tomu nebude na očakávanej úrovni. Ak však firma vhodne využije nástroje a postupy pre riadenie kvality a popri tom využíva testovanie počas celého vývoja projektu, musí síce vynaložiť väčšie počiatkové náklady, tie sa jej však mnohonásobne vrátia v podobe menšieho počtu chýb objavených zákazníkom, nižších nákladoch na opravu chýb a v konečnom dôsledku väčšej spokojnosti klientov firmy. Toto je nakoniec významný faktor v konkurenčnom prostredí pri získavaní zákaziek.

Použitá literatúra

1. Bieliková, M.: *Softvérové inžinierstvo - Princípy a manažment*. s.l. : Vydavateľstvo STU, Bratislava
2. Kroeger T., Davidson N.: A Perspective-based Model of Quality for Software Engineering Processes. 2009. Dostupné na internete : [Online] [Dátum: 14. 10 2009.] <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5076637&isnumber=5076603>
3. Patton, R.: *Testování softwaru*. Praha : Computer Press,a.s, 2002. (angličtina)
4. Schwalbe, K.: *Řízení projektu v IT*. Brno : Computer Press, a.s, 2007. (angličtina)
5. Tanuška P., Schreiber P.: Poznámky k problematike testovania informačných systémov. *atpjournal*. (1999). Dostupné na internete : [Online] [Dátum: 3. Október 2009.] http://www.atpjournal.sk/casopisy/atp_99/atp10/tanuska.htm

Annotation

Can Testing save Quality?

Nowadays, in the age of fierce competition and conscious clients is Quality an often repeated word at many meetings. The situation in software section is similar. Companies try to develop a quality software to satisfy customers' needs, gain a good reputation and adequate profit and also minimize the costs to troubleshoot low-quality software. The term 'quality software' is not easy to define and there are many ways to describe it. In my opinion testing shouldn't be the last step in the process of development of a quality product. Therefore this essay focuses on the relation of quality and testing. I will try to specify the term 'quality software' and closely describe the importance of testing in the process of software development. This specification should limit unrealistic expectations and inadequate implementation of testing into the product development.