

VNÚTORNÁ KVALITA SOFTVÉRU A JEJ VPLYV NA SOFTVÉROVÉ PROJEKTY

„Vnútoraná kvalita softvéru je aspekt vývoja, ktorý majú manažéri najmenej na očiach.“

Bc. Ján Kmeťko

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

Abstrakt. *Ked' sa povie riadenie kvality softvéru, väčšina ľudí si vybaví dôkladné otestovanie jeho funkčnosti. Často zanedbávaná však býva kontrola vnútornej kvality softvéru - kvalita návrhu, zdrojového kódu a dokumentácie. Zákazníci často takýto stav nepostrehnú, lebo nejde o aspekty, ktorými by sa softvér prejavoval navonok. Dodávateľ upustením od vnútornej kvality zníži svoje náklady na vytvorenie prvej verzie a navyše viac upevní svoju pozíciu u zákazníka (u projektov na mieru), lebo iná firma nebude schopná uskutočniť zmeny na dodanom softvéri. Za zložitejšie úpravy v budúcnosti navyše môže účtovať viac peňazí, lebo zákazník nemá na výber iného dodávateľa a iný dodávateľ by pýtal za úpravy ešte viac. Aký je vplyv vnútornej kvality softvéru na všetky zúčastnené strany v softvérovom projekte? V eseji sa autor zamýšľa aké teoretické, praktické aj ekonomické príčiny môžu vplývať na zníženie vnútornej kvality softvérových projektov a aké sú dôsledky takéhoto zníženia.*

Kľúčové slová: *vnútorná kvalita softvéru, kvalita softvéru*

Úvod

Už v začiatkoch éry počítačov bolo nevyhnutné kontrolovať správnu funkčnosť vytvoreného hardvéru a softvéru. Spôsob kontroly funkčnosti softvéru je však od iných oblastí priemyslu špecifický a výnimočný. Výnimočnosť spočíva v zložitosti softvéru a tiež v jeho ľahkej reprodukovateľnosti. Takže náklady na výrobu softvéru sú neporovnateľne vyššie, ako náklady na jeho rozmnoženie.

Vyrovnanie sa s vnútornou zložitou softvéru a jej zvládnutie je jedným zo základných cieľov softvérového inžinierstva. Reálny softvér je taký zložitý, že matematicky, alebo teoreticky dokázať jeho správnosť je prakticky nemožné.

Vytvorením prvej verzie úspešný softvérový projekt nekončí. Vytvorený softvér je potrebné udržiavať, opravovať chyby a pridávať novú funkcionality. Zatiaľ čo zákazník vníma kvalitu softvéru len cez jeho vonkajšie prejavy (funkčnosť, rýchlosť odozvy, atď.), programátori a vývojári, ktorí majú softvér udržiavať ho vnímajú cez jeho vnútornú kvalitu – cez to, ako je navrhnutý, napísaný a zdokumentovaný. Zaujímavé sú aj ekonomické aspekty vnútornej kvality, lebo jej zanedbanie síce prinesie rýchlejší okamžitý posun vpred, ale o to viac môže skomplikovať posun projektu vpred v budúcnosti.

Vnútorná kvalita softvéru

Pod vnútornú kvalitu softvéru podľa normy ISO 9126 zaraďujeme tie jeho aspekty, ktoré nie sú viditeľné používateľom [1]. Napriek tomu sa však tieto aspekty podieľajú na charakteristikách ako rozšíriteľnosť, prispôsobiteľnosť a znovupoužiteľnosť.

Vnútorne kvalitný softvér má dobrý návrh, dobre a prehľadne zapísaný kód, kód je dobre okomentovaný a softvér je celkovo dobre zdokumentovaný. Vnútorne kvalitný softvér je zrozumiteľný a rýchlo pochopiteľný. Problémom je, že vnútornú kvalitu je veľmi ťažké merať. Aj keď existujú nástroje, ktoré napríklad vypočítajú zložitost kódu (vnorenia a cykly), takéto metriky nemusia exaktne ukazovať na zlý, alebo dobrý kód. Posúdenie vnútornej kvality je o cite a skúsenostiach experta, ktorý ju posudzuje a nedá sa zatiaľ exaktne merať [4]. Tiež by nemalo ísť o experta, ktorý sa podieľal na vývoji, lebo ten je ovplyvnený jeho účasťou na projekte a má o projekte také poznatky, ktoré mu ho pomôžu ľahšie pochopiť aj keď je vnútorne nekvalitný. Preto by mal vnútornú kvalitu posudzovať nezaujatý expert, ktorý sa vyzná v použitých technológiách a problémovej doméne, ale projektu sa priamo nezúčastnil [4]. Podľa môjho názoru by mal byť každý projekt tak dobre napísaný a zdokumentovaný, aby sa v ňom expert spĺňajúci vyššie popísané predpoklady vedel v adekvátnom čase zorientovať a zodpovedne uskutočniť prípadné zmeny.

Ako dosiahnuť vnútornú kvalitu

Vnútnú kvalitu je možné dosiahnuť len tak, že ju bude mať na pamäti celý tím počas celého vývoja [2]. To znamená, že na ňu musí myslieť nielen programátor a návrhár, ktorí softvér tvoria, ale aj manažér, ktorý ich svojimi rozhodnutiami nesmie tlačiť do jej znižovania. Nie je vhodné príliš tlačiť na zrýchlenie vývoja, lebo vtedy budú mať realizátori na pamäti rýchlosť postupu a nie kvalitu, čo následne vyústi do zníženej kvality. Pre programátorov je veľmi lákavé robiť veľké skoky vpred rýchlym pridávaním funkcionality, ale ak zabudnú na dokumentáciu alebo dobrý návrh, tieto skoky vpred sa stanú kontraproduktívne. Potom je možnou cestou zlepšenia vnútornej kvality dodatočný refaktoring kódu a návrhu.

Vnútnú kvalitu treba mať stále na pamäti a aj ju priebežne kontrolovať. Podľa môjho názoru najlepším mechanizmom je skupinová revízia návrhu aj kódu. Pri nej autor vysvetľuje ostatným členom tímu kód, alebo návrh a ostatní v ňom hľadajú chyby.

Výhodou je, že kód bude poznať viacero ľudí a lepšie sa v tíme šíria skúsenosti, vedomosti a informácie. Kontrolu a udržiavanie vnútornej kvality by mala priamo podporovať použitá metodika vývoja softvéru. Tá musí byť nastavená tak, aby brala do úvahy skúsenosti tímu celkové, ale aj s použitými technológiami. Metodika by mala dostatočne definovať úroveň dokumentácie, ako aj predpisovať revízie kľúčových častí analýzy, návrhu a kódu.

Zaujímavá je aj súvislosť medzi vnútornou a vonkajšou kvalitou. Na jednej strane dobre napísaný a zdokumentovaný kód by mal obsahovať menej chýb, takže zlepšenie vnútornej kvality počas vývoja by sa malo prejavíť zvýšením vonkajšej kvality prvej verzie. Na strane druhej neustále zmeny v kóde, ktoré sú počas údržby realizované za účelom zvýšenia vonkajšej kvality, majú často veľmi negatívny dopad na kvalitu vnútornú. Jednoducho sa s niektorým typom zmien nerátalo, keď sa písal pôvodný kód a tak je jednoduchšie zmenu „nejako“ uskutočniť a prižmúriť oči nad pokazením kvality kódu. Podobne ak nie ešte viac to platí o zmenách v návrhu. Meniace sa prostredie a zmeny majú teda na softvér degeneračné účinky. A práve postupne upadajúca vnútrotná kvalita softvéru je dôvod, prečo aj pôvodne dobre napísaný softvér je potrebné po čase napísať kompletne od nuly [3].

Iné možnosti zvýšenia vnútornej kvality

Moderné prístupy k vývoju, ako napríklad modelom riadený vývoj softvéru, môžu výrazne zvýšiť vnútornú kvalitu. Je jednoduchšie udržať kvalitný model, ako udržať kvalitný kód. Ak je model interpretovaný, k vygenerovaniu kódu vôbec nedôjde. Ak sa použije generovanie kódu, je jeho výsledná kvalita závislá od kvality šablón, ktoré sa použili na generovanie. Napríklad pri vývoji štandardných intranetových aplikácií je s použitím dobrého nástroja možné generovať viac ako 90% výsledného kódu. Rôznymi technikami vkladania nevyhnutného kódu do modelu sa dá toto číslo dokonca zvýšiť na 100%. Bola by preto veľká škoda generovať nekvalitný, nezrozumiteľný, alebo neokomentovaný kód.

Modelom riadený vývoj softvéru tak ponúka možnosť výrazného zníženia objemu udržiavaných artefaktov (udržiavame iba rozsahovo menší model namiesto kódu), čím znižuje celkovú zložitosť softvéru.

Ďalšou možnosťou dodatočného zvýšenia vnútornej kvality je refaktoring. Ide o zvyšovanie kvality už napísaného kódu pri súčasnom zachovaní vonkajšieho správania. Pustiť sa do rozsiahlejšieho refaktoringu však môže byť riskantné, lebo sa ním môžu do softvéru zanesť nové chyby a vyžiada si zvýšené náklady. Náklady na refaktoring sú spravidla vyššie, ako náklady na vytvorenie dobrého kódu od začiatku, lebo programátori musia opäť porozumieť princípom, ako kód vnútorne funguje. Z vlastných skúseností viem, že vylepšiť komplikovaný a nezdokumentovaný kód je niekedy nemožné a lacnejšie by bolo napísať celý modul od začiatku.

Ekonomické súvislosti

Zaujímavé sú aj ekonomické súvislosti vnútornej kvality. Investícia do vnútornej kvality je investícia dlhodobá. Dôraz na vnútornú kvalitu pravdepodobne zvýši náklady na vytvorenie prvej verzie, lebo sa viac reviduje, refaktoruje a dokumentuje. Je to hlavne

investícia do zefektívnenia úprav, ktoré budú uskutočnené v neskoršej budúcnosti, možno aj inými programátormi a v inej firme.

U „krabicových“ softvérov, ktoré sa predávajú veľkému počtu zákazníkov, je ekonomické rozhodnutie zväčša jednoduché. Čím je softvér kvalitnejšie vytvorený, tým sa pružnejšie sa doň zapracovávajú zmeny a tým sa softvér stáva konkurencie schopnejším. Zákazníci si kupujú konkrétny produkt tak, ako je, a ťarcha zmien, aktualizácií a údržby (riziko) je na strane dodávateľa. Ak je zákazník veľmi nespokojný, prejde na konkurenčný produkt. Pokiaľ softvér nemusí byť uvedený do konkrétneho termínu (napríklad dátum platnosti nového zákona), je výhodnejšie vytvoriť softvérový produkt s dôrazom na kvalitu a ušetriť si tak náklady na jeho budúcu údržbu, ako narychlo vytvoriť softvér, čo síce spĺňa požiadavky, ale pod ťarchou zmien zdegeneruje omnoho rýchlejšie ako dobre navrhnutý a vytvorený produkt.

Úplne iná situácia však nastáva u softvérov vytváraných na mieru. Tam nesie väčšinou projektové riziko zákazník. Máloktorá softvérová firma vezme na seba dobrovoľne riziko projektu, keď ho môže nenápadne presunúť na zákazníka. Na Slovensku sa kontrakty na mieru nezvyknú uzatvárať na celé obdobie životnosti softvéru. Nezvykne sa ani zmluvne špecifikovať požadovaná vnútorná kvalita softvéru, lebo, ako už bolo spomenuté, je veľmi obtiažne ju merať. Predmetom zmluvy je potom väčšinou len dodanie softvéru, ktorý spĺňa funkcionality špecifikované v prílohe zmluvy (časť analýzy). Chybovosť sa zvykne vyriešiť tak, že chyby, ktoré zákazník objaví v určitom časovom horizonte, sú opravené bezplatne. A práve tu úmyselne, alebo neúmyselne vzniká jedna zvrátená možnosť zneužitia nevedomosti zákazníka.

- Vytvoriť prvú verziu vnútorne menej kvalitného softvéru by malo byť lacnejšie.
- Ťarcha budúcich úprav je na zákazníkovi.
- Máloktorý zákazník má expertov schopných posúdiť vnútornú kvalitu dodaného softvéru. Väčšinou vnútornú kvalitu nevidí, a často ho ani nezaujíma.
- Ak by chcel zákazník zmeniť dodávateľa, má to ťažšie, lebo konkurencia si za úpravy nekvalitného, zle zdokumentovaného softvéru musí zákonite vypýtať viac peňazí.

Nie je potom lákavé pre dodávateľa upustiť od vnútornej kvality a zvýšiť si tak zisk? A nemusí to byť len úmysel. Stačí, ak si pracovníci obchodných oddelení a manažéri zvyknú, že skracovanie termínov a škrtenie prác na dokumentácii zákazníkovi nevadí, práve naopak, skôr dodaný softvér je zvyčajne v očiach zákazníka plusom. Alebo stačí aj to, že sa projekt dostane do časového, alebo rozpočtového sklzu. Vtedy je tiež najľahšie obetovať vnútornú kvalitu.

Napriek tomu, že väčšina publikácií a článkov vyzdvihuje vnútornú kvalitu softvéru, moje skúsenosti z praxe na Slovensku sú skôr opačné. Väčšina projektov, na ktorých som pôsobil, nebola dobre zdokumentovaná a do písania kvalitného a zdokumentovaného kódu nikto mňa ani ostatných členov tímu netlačil. Vnútorná kvalita kódu nebola skoro vôbec kontrolovaná. Preto kvalita kódu ostávala na len na rozhodnutí a vóli programátorov.

Členovia vývojového tímu

Skúsme sa zamyslieť, aké má dopady vnúťorná kvalita na jednotlivých členov vývojového tímu a aké dopady majú členovia tímu na vnúťornú kvalitu.

Projektový manažér

Vnúťorná kvalita softvéru je aspekt vývoja, ktorý majú manažéri najmenej na očiach. Ak sa dostane projekt z dôvodu zlej vnútornej kvality do problémov, málokedy je z toho vinený priamo projektový manažér. Podľa môjho názoru je to chyba, lebo aj keď nemá na vnúťornú kvalitu priamy dosah, jeho rozhodnutia vnúťornú kvalitu výrazne ovplyvňujú. Pre manažéra je vnúťorná kvalita často skrytým aspektom softvéru. Táto pozícia býva podľa mojich skúseností často obsadzovaná ľuďmi bez technického vzdelania, následkom čoho nerozumejú do detailov procesu vývoja softvéru a jeho súvislostiam. To, že je manažérom vnúťorná kvalita skrytá, mylne evokuje, že ju nemôžu ovplyvniť. Na projektových poradách vysielajú manažér signály ostatným členom tímu. Ak bude tlačíť na programátorov, aby nejakú úlohu spravili rýchlejšie, pravdepodobne tým zníži vnúťornú kvalitu výsledku, lebo programátori úlohu pod časovým stresom iba narýchlo „zbúchajú“. Ak naopak bude na vnútornej kvalite trvať a poverí jej kontrolou správnych ľudí, pravdepodobne do bude mať na ňu pozitívny dosah. Preto by si podľa mňa mal každý projektový manažér uvedomovať tieto súvislosti a správne vyrovnať kvalitu s nákladmi. Projektoví manažéri, ktorí si tieto súvislosti uvedomujú a konajú v súlade s nimi, by mali byť príkladom pre ostatných v rovnakej funkcii.

Programátor, návrhár, dokumentarista

Úplne v inej pozícii sú voči vnútornej kvalite realizačné roly. Ich práca má bezprostredný vplyv na vnúťornú kvalitu softvéru. To, ako je kód dobre napísaný a ako dobre je zdokumentovaný, závisí od programátora. Kvalita návrhu závisí od návrhára a kvalita ostatnej dokumentácie od dokumentaristu. Ak sa v projekte ukáže ako problém nízka vnúťorná kvalita, bude vina pravdepodobne prisúdená realizátorom.

Zaujímavé je, že autori softvéru s nízkou vnúťornou kvalitou tento problém často nevnímajú [4]. Je to tak zrejme z toho dôvodu, že nimi napísaný kód alebo vytvorený návrh im je dôverne známy a preto nepociťujú, ak je riešenie neprehľadné, neintuitívne, alebo zle zdokumentované. Jedným zo závažných paradoxov je, že autor nezdokumentovaného a nekvalitného kódu zvyšuje svoju dôležitosť pre projekt a firmu. Ako jediný totiž dôverne pozná daný kód a ktokoľvek iný by chcel doň preniknúť, musí vynaložiť neprimerané úsilie. V niektorých prípadoch dokonca môže vyzeráť ako neschopný ten, čo má upravovať nekvalitný kód namiesto toho, aby bol na zodpovednosť vzatý autor nekvalitného kódu. Autor zlého kódu tiež riskuje zhoršenie svojej povesti u programátorov, ktorí po ňom majú neprehľadný kód prerábať a upravovať. Väčšinou sa ale môže vyhovoriť, že bol do nízkej kvality dotlačený potrebou splniť termíny, respektíve opak mu je po čase ťažké dokázať. Počas svojej praxe som videl už veľa neprehľadných kódov a o to viac si vážim, keď sa mi dostane rúk kvalitná práca.

Malo by byť cnosťou programátorov písať kvalitný a zdokumentovaný kód, za ktorý sa nebude musieť hanbiť, keď ho po ňom bude upravovať niektorý jeho kolega. Ak však nikoho okrem programátora rozdiel (v čase tvorby kódu) nezaujímá, je prirodzenou

ľudskou vlastnosťou vybrať si ľahšiu cestu a robiť rýchlo a nekvalitne. Ja sám som často využil rýchlejšiu cestu pri niekoľkých školských projektoch, o ktorých som vedel, že pôjdu hneď po odprezentovaní funkčnosti do zabudnutia a už ich nikdy nepoužijem. Hlavne u realizačných rol platí, že na dosiahnutie vnútornej kvality je potrebné neustále sa sústreďovať na ňu. Dobrý a kvalitný kód totiž skôr evokuje kvalitu udržať, kým nekvalitný kód sa málokto podujme z vlastnej iniciatívy zlepšiť a programátor má skôr tendenciu ešte viac kvalitu znížiť.

Rada pre zákazníkov

Či má pre dodávateľa softvéru zníženie vnútornej kvality pozitívny alebo negatívny efekt, je veľmi diskutabilné (viď. ekonomické súvislosti) a závisí to od konkrétneho projektu. To, že zlá vnútorná kvalita je pre zákazníkov veľké negatívum, je fakt. Fakt, ktorý sa opiera o vyššie náklady na úpravy vo vzdialenejšej budúcnosti, ktoré bude pravdepodobne znášať zákazník. Fakt, ktorý sa tiež opiera o to, že softvérové firmy si často účtujú také sumy, ako keby dodávali najkvalitnejší softvér na svete.

Pred dvomi rokmi som robil na testovacom oddelení v nemenovanej slovenskej banke. Celé oddelenie s počtom cca 50 ľudí vyše roka testovalo funkčnosť dodávaného softvéru. Mojou úlohou bolo kontrolovať či sú do vyvíjaného systému správne importované dáta. Tie sa doň každý víkend importovali z iných systémov. Na starosti to mal jeden veľký databázový skript napísaný v SQL s rozsahom približne štyridsať strán A4. Jednotlivé SQL príkazy boli dlhé viac ako dve strany, pričom nasledujúci príkaz závisel od toho predchádzajúceho. Celý skript neobsahoval žiadne komentáre. Navyše predmet importu (čo všetko a ako sa má importovať) nebol poriadne a jednoznačne v žiadnom dokumente špecifikovaný. Nečudo, že moja snaha otestovať import dát neskončila úspešne a do skončenia mojej brigády sa mi to nepodarilo. Iste si viete predstaviť, ako ťažko by sa robili zmeny v tomto skripte, v prípade, že by jeho autor z projektu odišiel. Som toho názoru, že od renomovanej softvérovej firmy mal zákazník rozhodne dostať omnoho kvalitnejší kód. A táto moja skúsenosť je veľmi dobrou ukážkou, že ani 50 členný testovací tím, ktorý sa zameriava na vonkajšiu kvalitu nezaručí zákazníkovi vnútornú kvalitný softvér.

Ako sa potom majú zákazníci brániť, aby dostali za svoje peniaze kvalitný produkt? Opäť platí už spomínaná múdrosť, že vnútornú kvalitu treba mať na pamäti počas celého projektu. Tejto múdrosti by sa mal držať aj zákazník a pokúsiť sa nejako zakotviť vnútornú kvalitu do dodávateľskej zmluvy. Väčšina zákazníkov však nemá ľudí, ktorí sa dostatočne rozumejú do tvorby softvéru, preto by zákazník mal získať na svoju stranu nezávislého experta, ktorý je schopný kvalitu posúdiť. Tento expert musí byť nezávislý a nekorumpovateľný. Tento expert by mal mať v rukách také možnosti, aby na určitej úrovni kvality mohol trvať. Zmluva by mala definovať mieru dokumentácie, ako aj podmienku, že dodaný produkt je možné odmietnuť na základe nízkej kvality kódu alebo návrhu a požadovať refaktoring. Expert na strane zákazníka by sa mal zúčastňovať revízií architektúry, kódu a návrhu a počas celého projektu tlačil na vysokú úroveň vnútornej kvality. Jeho úlohou by bolo zabrániť manažérskym rozhodnutiam, ktoré by znížili kvalitu (neúmerné skracovanie termínov a znižovanie prostriedkov) a dohliadať nad dodržiavaním takých procesov, ktoré kvalitu zabezpečia (dodržanie metodiky a

štandardov). Myslím si, že je pre zákazníka veľmi dobrou investíciou zabezpečiť si takéhoto experta dohliadajúceho na projekt.

Záver

Cieľom eseje bola analýza širších súvislostí medzi vnútornou kvalitou softvéru a jej vplyvom na softvérové projekty. Analyzoval som vplyv jednotlivých rol pri vývoji softvéru na vnútornú kvalitu, ako aj jej ekonomické súvislosti. Docieliť vysokú vnútornú kvalitu je možné len neustálym sústredením sa na ňu každým členom tímu. Ide o aspekt, ktorý treba mať na pamäti a od vnútornej kvality by sa malo upúšťať len cieľavedome a uvedomujúc si dopady.

Dostatočná vnútorná kvalita je najmä v záujme zákazníka, aj keď zodpovedné osoby na jeho strane si nedostatok vnútornej kvality často nevšimnú, pre nedostatok ich skúseností s tvorbou softvéru. Preto by mal zákazník pri každom väčšom projekte získať do projektu experta, ktorý bude hájiť jeho záujmy a dohliadať na vnútornú kvalitu softvéru.

Použitá literatúra

1. Ho-Won Jung, Seung-Gweon Kim, Chang-Shin Chung: *Measuring Software Product Quality: A Survey of ISO/IEC 9126*. IEEE Software, vol. 21, no. 5, 88-92, Sep/Oct, 2004.
2. McConnell, S.: *Code Complete, 2nd Ed.*, Microsoft Press, 2004
3. Paleta, P.: *Co programátori ve škole neucí*, Computer Press, 2003
4. Plosch, R. (2007). The EMISQ Method - Expert Based Evaluation of Internal Software Quality. *31st IEEE Software Engineering Workshop*, (pp. 99-108).

Annotation

Internal software quality and its influence on software projects

When talking about software quality most people think that it's all about careful testing of its functionality. But on the other side, internal quality, quality of design, code and documentation is very often underestimated. The customers would not detect bad internal quality, because it is not an aspect of software which they would be able to see from outside. Software companies can decrease their expenses for creating the first version of software by losing focus on internal quality. And moreover customer won't probably change contractor in the future, because competition would need to ask even higher price for future software changes, because undocumented software is more difficult to change for people who hasn't created it. And more complicated changes in the future also means higher amount of work and higher future sales. In the essay author thinks about theoretical, practical and economical reasons which can lead to low internal quality and what is the impact of low quality on all project stakeholders.