

ASPEKTY A MANAŽMENT KVALITY VYBRANÝCH DRUHOV APLIKÁCIÍ

*Kvalitný produkt vedie k spokojnosti zákazníka.
Manažment kvality zahŕňa všetky fázy jeho vývoja.*

Martin Lipták

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

Abstrakt. *Kvalita softvéru pozostáva z funkčných aspektov, ktoré zákazník vidí a štrukturálnych aspektov, ktoré zákazník vidí len v prípade ich nedostatkov. Kvalita musí byť zahrnutá do produktu v každej fáze vývoja, vrátane špecifikácie, pretože zle špecifikovaný produkt je nekoalitný produkt, pretože nemá hodnotu pre zákazníka. Naopak produkt, do ktorého bola kvalita zahrnutá v každej fáze jeho vývoja, vedie k spokojnosti zákazníka. Kvalita musí byť v procese neustále zlepšovaná. Esej porovnáva jednotlivé aspekty kvality webových, mobilných a stolových aplikácií.*

Kľúčové slová: *manažment kvality, kvalita softvéru, „jidoka“, webové aplikácie, mobilné aplikácie, stolové aplikácie*

Úvod

Kvalita softvéru môže byť chápaná z dvoch uhlov pohľadu. *Funkčný pohľad* rozumie kvalitu ako mieru spokojnosti zákazníka. Zahŕňa aspekty ako výber funkcií produktu, vzhľad produktu a jeho použiteľnosť (usability), ktoré si zákazník všimne ako prvé. *Štrukturálny pohľad* ukazuje produkt zvnútra a usiluje sa o nízku chybovosť, vysokú stabilitu, jednoduchú údržbu, rozšíriteľnosť, rýchlosť a bezpečnosť. Ide o veci, ktoré zákazník vo väčšine prípadov priamo nevidí, ale sú nevyhnutné pre používanie produktu. Štrukturálna kvalita sa zabezpečuje testovaním, kontinuálnou integráciou (continuous integration), refaktorovaním kódu a tvorbou primeranej dokumentácie pre budúce

rozširovanie produktu. V eseji by som rád porovnal jednotlivé aspekty kvality pri vývoji webových, mobilných a stolových (desktop) aplikácií.

Manažment kvality

Manažment kvality sa vykonáva v každej časti vývoja softvéru. Kvalita musí byť od začiatku zakomponovaná do špecifikácie, návrhu, implementácie, testovania, nasadenia a údržby produktu a neustále v procese zlepšovania. V prípade, že by sme sa o kvalitu starali až tesne pred nasadením, pravdepodobne zistíme, že softvér má vážne nedostatky, ktoré v danej chvíli už nie je možné napraviť. Tu sa ukazujú výhody prístupu lean management a agilných metód vývoja. Neustále zlepšovanie kvality sa nazýva aj ako „jidoka“ a je jednou z dvoch základných častí lean management. Druhou časťou je „muda“, ktorá označuje odstraňovanie častí procesu vývoja, ktorý neprináša hodnotu pre zákazníka. Obidve časti sú dôležité pre správne fungovanie prístupu lean management a agilných metód vývoja, ktoré tento prístup realizujú na nižšej úrovni vývoja produktu. [2]

Funkčná kvalita

Funkčná kvalita sa niekde označuje ako miera splnenia požiadaviek zákazníka. V prípade *softvéru* COTS (commercial off-the-shelf) by som chcel kvalitu v kontexte trhu poňať z ešte širšej perspektívy. Kvalita je vo väčšine prípadov podmienkou presadenia sa nového produktu na trhu, kde nestačí produkt vyvinutý podľa špecifikácie jedného zákazníka, ale treba robiť kompromisy, aby boli s produktom spokojní viacerí zákazníci. V prípade *zákazníckeho softvéru*, kde je produkt vyvinutý spravidla podľa požiadaviek jedného zákazníka, je tiež dobré myslieť na budúce využitie častí softvéru v ďalších produktoch alebo rozšírenie softvéru na COTS verziu. Už samotná špecifikácia musí byť tvorená s vysokým ohľadom na kvalitu budúceho produktu a musí čo najlepšie odrážať skutočné potreby zákazníka. Produkt, ktorý dokonale spĺňa nekvalitnú špecifikáciu, nemôže byť kvalitný. Akým spôsobom vyvinúť kvalitnú špecifikáciu v prostredí webových, mobilných a stolových aplikácií?

Webové a mobilné prostredie prirodzene navádzajú na agilné metódy vývoja. Je to dané potrebou rýchlych zmien, veľakrát neistotou trhu (ak vytvárame niečo nové) a vyššou toleranciou chýb, keďže nejde o aplikácie, ktoré riadia jadrovú elektrárňu [3]. Agilné metódy umožňujú vykonať viac iterácií vývoja a tým softvér lepšie prispôbiť potrebám zákazníka. Ak vytvárame produkt zameraný na trh, či už ako platenú webovú aplikáciu alebo komerčnú mobilnú aplikáciu, máme na začiatku iba nápad a veľmi približný a subjektívny odhad trhu. Tým, že vykonáme iba malú iteráciu so základnou funkčnosťou, ktorá vyjadří zmysel aplikácie, môžeme okamžite dostať spätnú väzbu od zákazníkov, znovu špecifikovať časti, ktoré sa ukázali ako nesprávne špecifikované (nevyhovujúce pre zákazníka znamená v prípade trhu nekvalitné) [4]. Oproti tomu, keby sme na začiatku implementovali celú aplikáciu, môžeme zistiť, že základná časť, na ktorej je postavené všetko ostatné, je nevyhovujúca pre zákazníka a vykonali sme zbytočnú prácu. Menej originálne webové a mobilné aplikácie, ktoré sa podobajú na iný projekt, na ktorom sme v minulosti pracovali (ide napríklad o elektronický obchod alebo redakčný systém), môžeme vytvoriť bez viacerých iterácií, pretože vieme vytvoriť správnu

špecifikáciu na začiatku a riziko, že sa produkt ukáže na konci ako nevyhovujúci je podstatne menšie vďaka našim predchádzajúcim skúsenostiam. V prípade stolových aplikácií je vývoj menej agilný a iterácia spravidla znamená novú verziu, ktorá musí pridať viac funkcií k tej predchádzajúcej, aby si zákazník kúpil aktualizáciu. Kvôli tomu je väčšina stolových aplikácií preplnená funkciami, ktoré používatelia nepoužívajú. Autori aplikácií sa často snažia získať spätnú väzbu odosielaním informácií o používaní produktu. Trend vývoja stolových aplikácií smeruje k agilným prístupom, ale táto oblasť je stále o niečo statickejšia ako v prípade mobilných alebo webových aplikácií.

Vzhľad je dôležitý pre zaujatie potenciálneho zákazníka a tiež spríjemňuje používanie aplikácie. Aplikácia, na ktorú sa používateľovi ťažko pozerá, má vážny nedostatok, ktorý musí vykompenzovať inými aspektmi kvality. Pri mobilných aplikáciách môže mať dizajn ešte väčšiu váhu, pretože používatelia sa na základe obrázkov aplikácie rozhodujú, či si danú aplikáciu vyskúšajú.

Použiteľnosť umožní používateľovi pracovať s aplikáciou intuitívne a jednoducho. Je to funkčná stránka vzhľadu, ktorá je síce dôležitejšia ako tá estetická, ale veľakrát s ňou súvisí. Aplikácia s funkčnosťou, ktorá sa ťažko používa, väčšinou nie je o nič lepšia ako aplikácia bez funkčnosti. Je dôležitá pre všetky typy aplikácií. [1]

Štruktúrna kvalita

Štruktúrna kvalita síce nie je na prvý pohľad viditeľná, ale je nevyhnutným predpokladom fungujúceho produktu a budúcej spokojnosti zákazníka. Základným spôsobom zaručenia štruktúrnej kvality je testovanie na viacerých úrovniach.

Najnižšou úrovňou sú *jednotkové testy* (unit tests) a integračné testy počas vývoja aplikácie. Najlepší spôsob tvorby testov je pred vývojom, čoho sa drží agilná metodológia *Test-driven development*. Takto je zaručené, že test skutočne testuje funkcionálnosť, ktorú chceme testovať. Testy musia byť písané priamo podľa špecifikácie používateľa (kvalitnej špecifikácie). Jednotkové testy testujú fungovanie jednotlivých vnútorných častí aplikácie. Veľakrát uľahčia samotný vývoj, pretože umožňujú vytvoriť tzv. *zástupcov* (*mocks* alebo *stubs*), ktorí nahradia iné jednotky aplikácie a danú jednotku je možné vyvíjať samostatne bez konfigurácie alebo dokonca vytvorenia ostatných jednotiek. *Integračné testy* otestujú funkčnosť celej aplikácie z hľadiska používateľa. V prípade, že vývojom pokážime určitú časť aplikácie, veľmi rýchlo sa dozvieme, že je niekde problém. Z vlastnej skúsenosti môžem povedať, že vo webovej aplikácii, kde som použil integračné testy sa objavuje podstatne menej chýb ako keď som vyvíjal bez nich. Integračné testy sa realizujú jednoducho v prípade webovej aplikácie, v prípade mobilných a stolových aplikácií je integračné testovanie ťažšie a často sa vykonáva „ručne“.

Na vyššej úrovni treba aplikáciu testovať priamo v prehliadačoch, na zariadeniach alebo operačných systémoch, kde sa bude používať. Webová aplikácia môže v inom prehliadači fungovať mierne odlišne, prípadne sa zobrazí inak. Dnes je situácia v oblasti dodržiavania štandardov lepšia ako v minulosti (Microsoft Internet Explorer 6 používa menej ako 1% používateľov). Nastupujúcim problémom sú rozdiely v používaní najnovších technológií HTML5, kde sa používajú prístupy *graceful degradation* alebo *progressive enhancement* [6]. *Graceful degradation* je prístup vytvorenia aplikácie s využitím všetkých možností nových technológií na začiatku a postupné ošetrovanie

prípadoch, keď tieto technológie nie sú dostupné. Napríklad aplikácia môže na prijímanie správ v reálnom čase využívať technológiu WebSockets, ktorá zníži záťaž klienta aj servera, ale v prípade nedostupnosti tejto technológie v prehliadači používateľa vykoná graceful degradation na klasický prístup periodickej kontroly prítomnosti nových správ na serveri (ajax polling). Progressive enhancement je opačný prístup s rovnakým cieľom. Na začiatku vytvoríme aplikáciu, ktorá používa iba zaužívané technológie, ktoré fungujú vo všetkých prípadoch a postupne aplikáciu rozširujeme o časti, ktoré využívajú najnovšie technológie a v prípade ich dostupnosti rozšíria alebo zlepšia funkčnosť aplikácie. Vo svete mobilných aplikácií je situácia komplikovanejšia. Existujú dve najpoužívanejšie platformy iOS a Android a veľké množstvo ďalších. Aplikácia vyvíjaná pre jednu platformu musí byť znova vyvinutá aj pre ďalšiu. V prípade platformy Android je situácia ešte horšia, pretože existuje veľa verzií operačného systému a veľa rôznorodých zariadení. Mobilná aplikácia musí s týmto všetkým počítať [5]. Tu sa tiež uplatňujú metódy *graceful degradation* alebo *progressive enhancement*. Situáciu mierne zlepšujú nástroje, ktoré umožnia vyvinúť aplikáciu naraz pre všetky platformy, tu sa ale stratia výhody tvorby aplikácie priamo pre cieľovú platformu. V prípade stolových aplikácií musí byť aplikácia testovaná pre všetky operačné systémy, pre ktoré je vyvíjaná.

Na najvyššej úrovni prebieha testovanie používateľom, kde sa ukáže do akej miery aplikácia spĺňa jeho požiadavky a či je kvalitná z hľadiska použiteľnosti. Testovanie použiteľnosti sa môže vykonávať v špecializovaných laboratóriách, kde sú sledované akcie používateľa. Takýmto spôsobom spoločnosť Canonical vykonáva pravidelne testovanie operačného systému Ubuntu, aby sa uistila, že prijaté rozhodnutia v rámci používateľského prostredia zvýšili použiteľnosť produktu [7].

Ďalším aspektom štrukturálnej kvality je kontinuálna integrácia. V prípade webových aplikácií je tento problém riešený nasadením na server, ideálne spôsobom, ktorý nespôsobí zobrazenie chýb používateľom, ktorí s aplikáciou práve pracujú. V prípade mobilných aplikácií znamená priebežná integrácia pravidelné dodávanie aktualizácií a nových verzií. Aktualizácia je často v rukách používateľa a vývojár má nižšiu kontrolu ako v prípade webových aplikácií. Pri stolových aplikáciách musí aktualizáciu často vykonať samotný používateľ, aj keď niektoré operačné systémy používajú tzv. balíčkovacie systémy (distribúcie GNU/Linux), ktoré riešia aktualizáciu všetkých aplikácií za používateľa.

Ďalším aspektom štrukturálnej kvality je dokumentácia. Jej rozsah závisí od veľkosti projektu, veľkosti tímu a od zamerania aplikácie. Pri malej aplikácii, na ktorej pracuje jeden vývojár, sú dostatočnou dokumentáciou vhodne pomenované jednotky zdrojového kódu a komentáre na miestach, kde môže komentár pomôcť ozrejmiť kód. Pri komentároch treba dávať pozor, aby bol aktuálny. V prípade, že zmeníme kód bez zmeny zodpovedajúceho komentára, riskujeme nedorozumenia v budúcnosti. Preto komentáre, ktoré nie sú nevyhnutné pre pochopenie kódu, poukazujú skôr na zlé praktiky (code smell) ako kvalitnú dokumentáciu. Pri väčšej aplikácii a väčšom tíme je dokumentácia dôležitá z hľadiska komunikácie v tíme a pre prípad výmeny členov tímu, kde môže pomôcť zorientovať sa novým členom. V prípade, že aplikácia obsahuje API pre iných vývojárov, dokumentácia API je nevyhnutná. Nezdokumentovaná funkcia API je takmer rovnaká ako neexistujúca funkcia API. Dokumentácia väčšinou nemá okamžité využitie v projekte, ktoré sa týka zákazníka, ale je dôležitá pre budúci vývoj a údržbu.

Je dôležité od začiatku myslieť na bezpečnosť. Treba si dať pozor, aby sme v záujme zvyšovania bezpečnosti neznížili použiteľnosť aplikácie (security by obscurity). Stupeň „bezpečnosti“ závisí od zamerania aplikácia. Internet banking môže vyžadovať zadanie veľmi silného hesla, ale aplikácia na ukladanie poznámok by nemala požadovať 3 nealfanumerické znaky v hesle (k takejto aplikácii má zmysel prihlasovanie cez Facebook alebo Twitter pre čo najvyššie pohodlie používateľa). Fenomén security by obscurity sa objavuje pri stolových aplikáciách aj v inej podobe. Používateľ je pri zakúpení väčšiny COTS produktov „otravovaný“ zadaním aktivačných kľúčov alebo aktiváciou cez Internet. Pre čestného používateľa je tento úkon často neprijemnou stratou času, prípadne sa môže stať, že používateľ, ktorý má softvér legálne zakúpený, stratil aktivačný kľúč a nemôže kvôli tomu softvér používať. Pre používateľa, ktorý má v úmysle používať softvér nelegálne, je táto ochrana neúčinná, pretože aktivačný kľúč môže nájsť na veľkom množstve stránok s kľúčmi k softvéru na Internete alebo môže pohľadať tzv. crack, ktorý odstráni ochranu z programu. [8]

Veľkosť a rýchlosť aplikácie by mali byť primerané jej funkcii a cielenému nasadeniu. Stolová aplikácia sa môže zmestiť na DVD, ale mobilná aplikácia samozrejme musí byť menšia a tiež by nemala sťahovať veľké objemy dát. V prípade webovej aplikácie sa tiež musíme zaoberať veľkosťou súborov, ktoré posielame webovému prehliadaču, pretože webová aplikácia môže byť používaná aj na mobilnom zariadení, kde vysoká rýchlosť pripojenia a výkonné spracovanie skriptov (pri náročnejších aplikáciách aj spracovanie skriptov vyžaduje nemalý výpočtový výkon) nie je samozrejmosťou.

Samozrejmosťou je primeraná rozšíriteľnosť a znovupoužiteľnosť kódu aplikácie. Pri webovej aplikácii nie je predpoklad znovupoužitia kódu taký vysoký ako mobilnej a stolovej aplikácii, kde vytvárame viac verzií aplikácie pre rôzne operačné systémy.

Záver

Sledovanie kvality webových, mobilných a stolových aplikácií má veľa spoločného, ale aj rozdielneho. Výrazným spoločným znakom je vysoko agilný prístup v prípade mobilných a webových aplikácií a postupný trend agilného vývoja v prípade stolových aplikácií. Takisto je zjavná snaha o štruktúrne aspekty kvality všetkých druhov aplikácií. Tieto aspekty som sa snažil čo najkomplexnejšie zhodnotiť v eseji.

Použitá literatúra

1. Bevan, N.: *Quality in Use: Meeting User Needs for Quality*. Elsevier, 1999.
2. Danovaro, E., Janes, A., Succi, G.: *Jidoka in software development*. In *Companion to the 23rd ACM SIGPLAN conference on Object oriented programming systems languages and applications - OOPSLA Companion '08*, ACM, 2008.
3. Feature, C.: *Using Risk to Balance Agile and Plan- Driven Methods*. IEEE, 2003, 57–66.
4. Gascoigne Goel: *Idea to paying customers in 7 weeks: how we did it*. 2011. <http://blog.bufferapp.com/idea-to-paying-customers-in-7-weeks-how-we-did-it>
5. Rahimian, V., Habibi, J.: *On Performance of Mobile Systems: Challenges of Software Design and Evaluation*.

6. Day S.: *Progressive enhancements or graceful degradation. Which are you?*. 2011. <http://www.simonday.com/web-design-blog/2011/03/24/progressive-enhancements-or-graceful-degradation-which-are-you/>
7. Sneddon J.: *User testing of Unity reveals some surprising results*. 2011. <http://www.omgubuntu.co.uk/2011/11/user-testing-of-unity-reveals-some-surprising-results>
8. Pavlovic D.: *Gaming security by obscurity*. ACM, 2011.

Annotation

Quality Aspects and Management of Selected Types of Applications

Software quality consists of functional aspects, that are visible to the customer, and structural aspects, that are visible only in cases of their insufficiency. Quality must be built into product in each development phase including specification. Incorrectly specified product is a bad-quality product, since it is of no value for the customer. On the other hand, product built with quality in mind in each one of its development phases, leads to customer satisfaction. Quality must be constantly improved in the development process. Essay compares aspects of quality of web, mobile and desktop applications.