

Zabezpečenie kvality v softvérovom projekte

TOMÁŠ ŠUREK

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

Abstrakt. V posledných rokoch sú softvérové produkty rozsiahlejšie a komplexnejšie. Komplexnosť má pritom kritický a strategický význam pri vyvíjaní softvérových produktov. Otázka kvality je v tomto prípade, keď diskutujeme o veľkosti a komplexnosti produktov, priamo kľúčová. Otázka znie, ako vytvoriť čo najkvalitnejší softvérový produkt pri obmedzenom čase a zdrojoch. Softvérové produkty sú nasadzované do prevádzky veľkých spoločností. Tieto spoločnosti pôsobia aj v kritických oblastiach ako zdravotníctvo a atómová energetika, kde prípadná chyba môže stáť životy ľudí. Na to, aby bola zabezpečená kvalita softvérových produktov je potrebné dodržiavať už dnes definované normy. Tieto normy však obsahujú iba odporúčania ako pri vývoji postupovať a nie ako softvér vytvoriť. V tejto eseji by som stručne opísal niektoré problematiky, ktoré sa riešia, a o ktorých sa diskutuje na Workshop on Software Quality (WoSQ). Cieľom práce je však opísať proces vývoja, testovania a vyhodnocovania kvality pri tvorbe softvérového produktu.

Úvod

Každý z nás sa stretáva s posudzovaním kvality denno-denne, či už v oblasti informačných technológií, alebo v ktorejkoľvek inej oblasti života. Z môjho pohľadu je rozdiel v tom, že kým pri posudzovaní kvality vecí z nášho bežného života to berieme akosi samozrejme v oblasti softvéru a softvérových produktov sme sa zamerali iba na funkčnosť a kvalitu sme pokladali za menej dôležitý atribút. Dnes je to už inak a kvalita už nie je iba jedným z atribútov softvérového produktu, je to vlastnosť, ktorá vytvára čoraz väčšiu konkurenčnú výhodu. Za vyššiu kvalitu sú dnes zákazníci ochotní platiť viac. O kvalite sa na medzinárodnej úrovni aj veľmi živo diskutuje a preto by som rád spomenul aké témy sa v súčasnosti riešia.

Teraz by som tu uviedol hlavné témy a ciele rozoberané na zatiaľ poslednom šiestom workshop on software quality[3]:

- Medzi-kultúrne otázky
- Hodnotenie softvérového produktu a certifikáty
- Definovanie softvérových procesov, vyhodnocovanie a zlepšovanie

Manažment projektov softvérových a informačných systémov, október 2008, s. 1-8.

- Kvalita softvérového vzdelávania
- Úvod do kvality softvérových programov
- Metódy a nástroje pre zabezpečenie kvality
- Softvér pre webový a objektovo - orientovaný vývoj
- Celkové riadenie kvality
- Budovanie kvality v softvérových produktoch
- Projektové riadenie a softvérová kvalita
- Spojenie kvality a rýchleho vývoja
- Testovanie, revízie, rekapitulácia a recenzie

Vymenovaním týchto tém som chcel naznačiť rozsiahlosť tejto témy a to, že kvalita sa dotýka všetkých častí vývoja softvérového produktu. Témou tejto eseje je pohľad na to, ako vytvoriť kvalitný softvérový produkt. Opíšem jednotlivé aspekty kvality a v druhej časti sa budem venovať testovaniu. Správne testovanie totiž patrí medzi nástroje potrebné k zabezpečeniu kvality výsledného produktu.

Softvérová kvalita

Riadenie kvality softvéru

Kvalitatívny štandard sa môže dosiahnuť iba s využitím efektívnych systémov na riadenie kvality. Kvalita je zabudovaná do softvérového produktu prostredníctvom riadiacich a technických postupov. Tie sú definované a vykonávané tak, aby bola zabezpečená[1]:

- Kvalita,
- Program a
- Plnenie rozpočtu

Medzi najdôležitejšie ciele softvérového inžinierstva patrí používanie technológií zlepšujúcich kvalitu softvéru. Niekoľko dôležitých technológií:

- Požiadavka na definíciu
- Prevencia porúch
- Detekcia chýb
- Odstránenie chýb.

Meranie softvérovej spoľahlivosti

Softvérovú spoľahlivosť možno merať, riadiť, a odhadnúť na základe historických údajov a vývoja. Zo štatistického hľadiska, softvérová spoľahlivosť je definovaná ako "pravdepodobnosť zlyhania určitých operácií v počítačovom programe v určitom prostredí, za určitú dobu".

Mnohí výskumníci sa pokúsili extrapolovať matematickú teóriu hardvérovej spoľahlivosti k predpovedi spoľahlivosti softvéru. Väčšina modelov hardvérovej spoľahlivosti je založených na zlyhaní v dôsledku opotrebenia, než zlyhanie v dôsledku konštrukčnej chyby. Bohužiaľ, opak je pravdou pre softvér. Faktom je, že všetky softvérové zlyhania môžeme vysledovať k návrhu alebo implementácii.

Ak vezmeme do úvahy základný počítačový systém, jednoduchým meraním je meranie spoľahlivosti - MTBF (priemerný čas medzi poruchami) a rovnako sa uvádza ako[1]:

$$- \text{MTBF} = \text{MTTF} + \text{MTTR}$$

kde

- MTTF: priemerná doba nastania poruchy
- MTTR: znamená čas na opravu

Okrem merania spoľahlivosti medzi, ďalšie merania patrí meranie dostupnosti. Ak si dostupnosť zdefinujeme ako pravdepodobnosť, že program je v súlade s požiadavkami na danom mieste v danom čase, môžeme dostupnosť definovať nasledovne:

$$\text{Dostupnosť} = \text{MTTF} / (\text{MTTF} + \text{MTTR}) \times 100\%$$

Myslím si, že tieto dva faktory spoľahlivosti a dostupnosť patria medzi základné ukazovatele pri posudzovaní kvality. Spoľahlivosť je v dobe, keď sú softvérové produkty nasadzované na 24 hodinovú prevádzku, braná veľmi zodpovedne. Ak by totiž produkt mal so spoľahlivosťou problém, stal by sa bezcenný. Dostupnosť podľa mňa nemá až taký kritický význam ako spoľahlivosť. Dostupnosť je totiž často-krát spojená s technickým riešením umiestnenia a prevádzky produktu. No chcel som ju zdôrazniť preto, že aj keď problém nie je v samotnom produkte to, že je nedostupný, sa automaticky prenáša do hodnotenia jeho kvality.

Zabezpečenie softvérovej kvality

Softvérová kvalita je zmapovanie riadiacich procesov a dizajn disciplín kontroly kvality na platné manažérske a technologické postupy softvérového inžinierstva. SQA (Software quality assurance) sú "zastrešujúce činnosti", ktoré sú použité pri každom kroku v procese tvorby softvéru. Uplatnenie SQA je problematické z dôvodu, že softvérová kvalita má zložitú povahu. Softvérová kvalita je jeden atribút počítačových programov, ktorý je definovaný ako konsenzus explicitne a implicitne zadaných požiadaviek. SQA zahŕňa nasledujúce činnosti[1]:

- Kvalita riadenia prístupu
- Efektívne softvérové inžinierstvo
- Formálne technické posudky, ktoré sú uplatňované v celom softvérovom procese
- Viac - stupňové testovacie stratégie
- Kontrola nad softvérovou dokumentáciou
- Postup pre zabezpečenie súladu s normami pre vývoj softvéru

- Meranie a mechanizmy pre podávanie správ.

Zabezpečovanie softvérovej bezpečnosti a analýza rizík sú činnosti, ktoré sa zameriavajú na identifikáciu a hodnotenie možných problémov, ktoré môžu negatívne ovplyvniť softvér a spôsobiť nevydarenie celého systému. Ak totiž existuje nebezpečenstvo, môže byť identifikované už na začiatku procesu. Najprv sa teda identifikujú nebezpečne časti a tie sa potom zoradia podľa kritickosti a riziká. Softvérový dizajn môže určiť či uvedené nebezpečenstvo bude odstránené, alebo sa bude iba kontrolovať.

ISO prístup k riadeniu softvérovej kvality

ISO 9000 opisuje kvalitu produktov, ktoré možno použiť pre akékoľvek podnikanie, bez ohľadu na výrobky alebo služby, ktoré ponúka. ISO 9000 zabezpečuje kvalitný model podniku zobrazujúci sieť vzájomne prepojených procesov. Pre kvalitný systém, ktorý má byť ISO-kompatibilný, sa tieto procesy musia zaoberať oblasťami, ktoré boli stanovené v norme. ISO 9000 opisuje jeden z mnoha prvkov systému zabezpečovania kvality vo všeobecnej rovine.

ISO 9001 je štandard kvality, ktorý sa vzťahuje na softvérové inžinierstvo. 9001/IS ISO 14001 je kvalitatívny model systému, ktorý je vhodný pre softvérový priemysel. Norma obsahuje 20 nasledovných požiadaviek[1]:

- Štatistické metódy
- Riadenie zodpovednosti
- Systém kvality
- Revízia zmluvy
- Dizajn kontroly
- Správa dokumentov a kontrola dát
- Nákup
- Ovládanie zákazníkom dodaného výrobku
- Identifikácia produktu a rozlišovacia schopnosť
- Proces kontroly
- Kontrola a skúšanie
- Riadenie inšpekcie, meracie a testovacie zariadenia
- Kontrolný a skúšobný stav
- Kontrola nevyhovujúcich výrobkov
- Nápravné a preventívne opatrenia
- Manipulácia, skladovanie, balenie, uchovanie a dodanie
- Kontrola kvality revízie
- Školenia
- Servis

9001/IS ISO 14001 je generické charakteru je žiaduce, aby bola táto norma vnímaná v kontexte softvérového priemyslu. V tomto smere bola poskytnutá v inom štandarde, konkrétne 9000-3/IS ISO 14000 (časť 3) , norma, ktorá vytvára väzbu medzi dvadsiatimi prvkami 9001/IS ISO 14001 a aktivitami v softvérovom priemysle. Povedal by som, že normy nám síce môžu pomôcť, ako by sme mali pri zabezpečovaní kvality postupovať, ale určite nám nedajú návod ako určitý softvérový produkt vyvinúť. Softvérové produkty sú rôznorodé a pri striktnom dodržiavaní noriem sa celý proces vývoja môže neúmerne predĺžiť alebo predražiť. Problém je v tom, že ak by sme to aj akceptovali nie je zaručené, že kvalita bude vyššia. Nakoniec by som dodal, že normy nás usmerňujú k tvorbe kvalitnejšieho softvéru. Zároveň si ale treba určiť prioritu jednotlivých bodov a podľa nej postupovať.

Testovanie

Keďže zložitosť aplikácií a softvéru sa zvyšuje, testovanie a hodnotenie sa stáva ťažším a jeho účinnosť je nižšia ako očakávania. Testovanie bolo často sledované čisto z technických dôvodov a počas posledných dvoch desaťročí došlo k výraznému pokroku v testovaní softvéru pomocou nových techník a metód. Základy testovania softvéru, sú[1]:

- Testovacie procesy, testovacie prípady a skúšobné plány
- Techniky, metódy, nástroje a štandardy
- Ľudia a organizácie.

Testovacie procesy, techniky a nástroje významne prispievajú k účinnému, efektívnemu testovaniu a zabezpečeniu kvality. Môžu však ponúknuť lepšie výsledky len vtedy, ak sú postavené na "ľudskom základe a morálnej manažérskej a organizačnej kultúre". Jedná sa o ľudí a kultúru v organizácii, ktorá určuje, ako bude akýkoľvek systém vykonávaný. Myslím si, že dobré medziľudské vzťahy, korektné správanie sa a dodržiavanie vhodných kultúrnych noriem je možno dôležitejšie, ako výber toho najoptimálnejšieho procesu, modelu či testovania. Všetky procesy, ich úspech alebo neúspech, totiž závisia od ľudí a tu musíme taktiež hľadať zlepšenie.

Modely vývoja a testovanie

V súčasnosti sa pri vývoji softvérových produktov využíva hlavne V-model, ktorý je založený na vodopádom modely. V-model nie je však dostatočne flexibilný a tak ho nahrádza nový model s názvom 4CC[2]. 4CC je moderný štruktúrovaný model poskytujúci efektívnejšiu tvorbu a testovanie softvéru.

Na to aby sme mohli hovoriť o testovaní je dôležité si špecifikovať pojem testovanie, keďže sa jeho význam mení. Tradičné testovanie znamenalo spustenie programu za účelom nájsť chyby. Novšia definícia by znela, že je to proces plánovania príprav, ohodnocovania cieľov a demonštrácia rozdielnosti aktuálneho a požadovaného stavu programu. Testovanie záleží aj od veľkosti firmy. Veľké firmy ako Microsoft majú samostatné oddelenie pre testovanie. Malé firmy si to nemôžu dovoliť, a preto je

testovanie integrované do vývojového procesu. Výber vývojového modelu, od ktorého závisí spôsob testovania, má na starosti projektový manažér. Úlohou testera je zase nájsť a oznamovať chyby a neskôr kontrolovať, či chyby boli odstránené. Buď sú chyby odstránené programátorom, alebo manažment môže rozhodnúť, že sa chyby nebudú odstraňovať.

Časom sa zistilo, že pri použití V-modelu vznikajú problémy. Prečo je V-model nedostatočný? V-model rozdeľuje testovanie na viacero úrovní. Začína z malými, už testovateľnými časťami, no postupne sa testujú väčšie celky. Pri testovaní väčších celkov však hrozí nebezpečenstvo zvýšeného počtu neodhalených chýb. Preto testovanie na každej úrovni musí byť veľmi dobre naplánované a kontrolované, aby sme sa vyhli prekryvaniu. Dva základné princípy V-modelu sú testovanie menších častí predtým ako sa použijú vo väčších celkoch a kvalitné plánovanie. Problémy pri V-modeli sú nasledovné[2]:

- Ignoruje sa fakt, že softvér je vyvíjaný v sérii produkčných verzií, kde každá ďalšia produkčná verzia mení správanie predošlej verzie.
- Nedokáže dostatočne pružne reagovať na zmeny

Four cycles of control(4CC) model

Pre potreby vývoja a kontroly softvérového produktu bol vytvorený 4CC model. Tento model poskytuje štruktúru, ktorá lepšie reaguje na zmeny. 4CC je vlastne vysokoúrovňový iteratívny a inkrementačný model, ktorý pozostáva zo štyroch časových cyklov[2]:

- Manažment stratégií vydávania
- Manažment vydávania projektu
- Manažment prírastkov
- Mini-míľniky

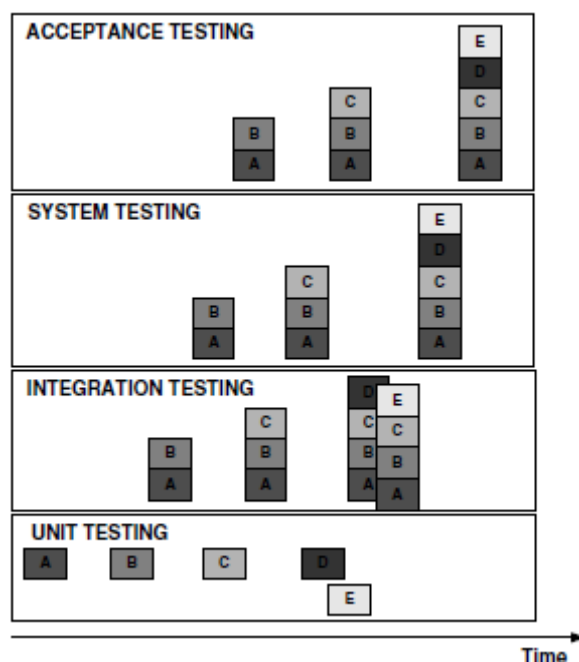
Každý z týchto štyroch časových cyklov poskytuje rôznu úroveň abstrakcie, aby uľahčil kontrolu a flexibilitu pri vývoji.

Testovanie začína už od prvej implementácie modulu. Na Obr. 1 je znázornené ako vlastne testovanie prebieha. Moduly A a B sú testované čiastkovo predtým ako ich spolu integrujeme. Integračné testovanie sa zameriava na kontrolu toho, či pridaný modul pracuje spolu so súčasnou verziou celého systému. V systémovom testovaní je celý systém overovaný so zameraním na celok, nielen na poslednú pridanú časť. Toto testovanie skúma produkt zo šiestich rôznych pohľadov[2]:

- Používateľského
- Medzinárodného
- Hardvérovej kompatibility
- Softvérovej kompatibility
- Splnenia špecifikácie
- Stability produktu

Akceptačné testovanie je typicky posledná úroveň testovania. Je to úroveň testovania, ktorá vlastne stále pretrváva, aj keď sa produkt dostane do prevádzky (používateľ sa stáva testerom). Úroveň akceptačného testu je zdôrazňovaná kvôli tomu, že zákazník kontroluje či produkt spĺňa jeho požiadavky. Akceptačný test pozostáva z dvoch častí[2]:

- Posúdenie z používateľského pohľadu (použiteľnosť, vhodnosť, aplikovateľnosť)
- Posledné kontroly pred vydaním produkčnej verzie



Obr. 1. Úrovně testovania pri inkrementálnom a iteratívnom vývoji[2]

Záver

Kvalita sa začína brať vážne na každom stupni vývoja softvérového produktu. Vznikajú preto rôzne automatizované procesy, do ktorých firmy vkladajú veľké finančné prostriedky. Táto esej sa zamerala na procesy a technológie potrebné pri tvorbe kvalitného softvéru. Sú opísané niektoré dôležité atribúty kvality, ako spoľahlivosť a dostupnosť. Spomenul som taktiež ISO prístup, ktorý nám pri správnom použití môže veľmi dobre poslúžiť. V druhej časti tejto eseje je rozobraná téma vývoja v testovaní a preskúmanie 4CC modelu, ktorý patrí medzi najnovšie vývojové modely. Ja som sa zameral hlavne na to, akým spôsobom sa vykonávajú testy v tomto modeli a čo nové prináša. Všetky časti eseje opisujú spôsob, akým by sa malo pri tvorbe kvalitného softvérového produktu postupovať. Dôležité však je uvedomiť si, že za

všetkým sú ľudia a ľudské rozhodnutia. Komunikácia a vnútorná kultúra firmy ďaleko viac ovplyvňujú kvalitu výsledného produktu ako si pripúšťame. Preto by som zhrnul zabezpečenie kvality do potreby kvalitného modelu, kvalitných procesov, kvalitných ľudí a ľudských vzťahov.

Použitá literatúra

1. Nasib S. Gill.: *Factors Affecting Effective Software Quality Management Revisited*, ACM SIGSOFT Software Engineering Notes, Vol. 30, No. 2 (March 2005).
2. Pyhäjärvi M., R. K., J. I.: *Increasing Understanding of the Modern Testing Perspective in Software Product Development Projects*, Hawaii International Conference on System Sciences (2002).
3. Sunita Chulani, B. B., J. V., B. W.: *Workshop on Software Quality*, Leipzig, Germany (2008).

Annotation

Assurance of software quality in project

In the last few years the software products are more extensive and more complex. During development of software products, complexity has critical and strategic importance. In this case of discussing sizes and complexity of products, the question of quality becomes the key question. The question is how to create the software product of highest quality in limited time with limited sources. Software products are integrated into services of big companies. These companies are involved in critical areas as health and atomic energy, where possible error can cause death of many people. Because we have to assure the quality of software products it is necessary to follow nowadays defined norms. These norms contain only recommendations how go through development process but not how to create software. In this essay I would shortly describe some problems, which are solving, discussing on Workshop on Software Quality (WoSQ). The aim of the work is to describe process of controlling, assuring quality and testing during development of software product.