

# Manažment kvality softvéru a jeho testovanie

OTO VOZÁR

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
oto\_vozar@zoznam.sk*

**Abstrakt.** Manažment kvality a testovanie majú v rámci inžinierstva a inžinierskeho prístupu dlhodobú tradíciu a sú s ním nerozlučne späté prakticky od jeho formálneho vzniku. Softvérové inžinierstvo ako jeden z najmladších inžinierskych oborov má prirodzenú potrebu zaradiť do svojich procesov podobné princípy, ktoré by zabezpečili kvalitu jeho výstupov prihladnuc na niektoré špecifiká, ktoré sú preň unikátne. Táto esej poskytuje definíciu oboch oblastí, hlavné ciele a princípy ich činnosti ako aj ich členenie na menšie procesy. V rámci manažmentu kvality sa potom zameriava na rôzne druhy stretnutí k projektu a prehliadkam kódu (code-reviews) a pri testovaní rozoberá rozdelenie testovacích techník a životný cyklus procesu testovania.

## Úvod

Kvalita je v dnešnej dobe často používaným slovom. Zákazníci požadujú čo najkvalitnejšie výrobky a firmy v snahe získať čo najviac zákazníkov investujú nemalé finančné prostriedky na to, aby im vyšli v ústrety. To isté platí aj pre oblasť softvérového inžinierstva a na kvalitu je kladený stále väčší dôraz, nakoľko sa trh softvérových produktov a služieb neustále rozširuje, rastie konkurencia medzi spoločnosťami a riešenia prenikajú do takých oblastí, kde od nich priamo závisia ľudské životy.

Existuje niekoľko rôznych definícií toho, čo predstavuje pojem kvalita, no všetky by sa dali zhrnúť do vety, že to je miera priblíženia sa k požiadavkám zákazníka. Ak teda bude cieľom vyrobiť nefunkčný softvér, bude tým kvalitnejší, čím menej bude fungovať [1]. Samozrejme, softvérové spoločnosti nastavujú svoje procesy skôr smerom k štandardným požiadavkám trhu medzi ktoré patrí okrem iných napríklad aj čo najnižšia chybovosť.

V súvislosti s danou témou sa budem zaoberať dvoma, myslím si, že najdôležitejšími činnosťami a to manažmentom kvality a testovaním. Kým prvý proces sa okrem statickej analýzy vlastností produktov zameriava aj na samotný proces ich

tvorby, druhý je viac zameraný na ich dynamickú analýzu a správanie sa v rôznych podmienkach.

## Manažment kvality

Manažment kvality softvéru sa vzťahuje na všetky aspekty softvérových procesov, produktov a zdrojov. Definuje procesy, ich majiteľov a požiadavky pre tieto procesy, merania týchto procesov a ich výstupov a dráhy pre spätné väzby [4]. Procesy manažmentu kvality pomáhajú zabezpečiť lepšiu kvalitu softvéru v danom projekte. Zároveň však poskytujú postrannú informáciu pre manažment indukujúcu kvalitu celkového inžinierskeho procesu.

Jednou z prvých fáz je plánovanie kvality, čo je proces, ktorý čiastočne patrí aj do celkového plánovania softvérového projektu. Plánovanie kvality v sebe zahŕňa hlavne:

1. Určenie požadovaných vlastností produktu v súvislosti s jeho charakteristikami kvality
2. Plánovanie procesov pre dosiahnutie takých vlastností.

Samotný manažment v sebe potom zahŕňa hlavne zabezpečenie kvality, verifikáciu a validáciu, prehliadky a audit.

Hlavnou úlohou procesu *zabezpečenia kvality* je plánovanie a vykonávanie aktivít pre získanie adekvátnej istoty, že je produkt vytváraný dostatočne kvalitne. V prvom rade sa je treba uistiť, že problém je jasne a dostatočne formulovaný a že požiadavky na riešenie sú presne definované a vyjadrené. Potom sa najčastejšie vypracuje plán, kde sa určia dokumenty, štandardy a praktiky týkajúce sa projektu a to ako bude kontrolované a monitorované ich dodržiavanie. Taktiež môže definovať techniky merania určitých vlastností, procedúry pre hlásenie problémov a opravné akcie, bezpečnosť fyzických médií a školenia.

*Verifikácia a validácia* sa zaoberá priamo kvalitou produktu a používa testovacie metódy pomocou ktorých môže lokalizovať jeho nedostatky. Zároveň jasne rozdeľuje zdroje, úlohy a zodpovednosti, takže potom je možné určiť, kto daný nedostatok spôsobil a mal by pracovať na jeho odstránení. Hlavný rozdiel medzi týmito dvoma procesmi je ten, že verifikácia sa snaží o to, aby bol produkt vytvorený správne v zmysle, že výstupy nejakej aktivity spĺňajú požiadavky určené predchádzajúcimi aktivitami. Na rozdiel od nej, validácia sa pokúša zabezpečiť, že produkt bude spĺňať zamýšľaný účel [4].

*Prehliadky a audit* sú podrobnejšie opísané v nasledujúcej kapitole.

## Prehliadky a audit

Tieto procesy v sebe zahŕňajú rôzne typy stretnutí v rámci projektu a môžeme ich rozdeliť na manažérske prehliadky, technické prehliadky, inšpekcie, previerky (walk-throughs) a audity.

*Manažérske prehliadky* sa zaoberajú viac procesom v rámci ktorého je vytváraný produkt než samotnými podrobnosťami jeho vývoja. To je dané mimo iného

aj tým, že manažéri často nie sú odborníkmi v technickej oblasti a skôr sa zaujímajú o to, či tvorba postupuje podľa plánovania, či nenastali nejaké problémy alebo zdržania a či sú správne a v dostatočnej miere pridelené všetky potrebné zdroje. Čo sa týka rozhodovania, väčšinou majú v kompetencii schvaľovať alebo odmietat' zmeny, ktoré by nejakým spôsobom zasiahli do výrobného procesu napríklad posunutím naplánovaných termínov, navýšením potrebných zdrojov alebo významnou zmenou vlastností vytváraného softvéru.

*Technické prehliadky* zasa naopak kladú dôraz na samotný produkt a jeho technické podrobnosti. Sú to formálne stretnutia, ktorých sa zúčastňujú prevažne samotní vývojári ale aj iní špecialisti z tých oblastí, ktorých sa vývoj dotýka. V rámci prehliadky musia byť vytvorené a pridelené špecifické roly a to vedúci stretnutia, osoba zodpovedná za rozhodnutia o danom produkte, tvorca zápisnice a ostatní členovia. Príkladom využitia takéhoto stretnutia môže byť situácia, keď má nejaký tím, ktorý je súčasťou väčšej firmy vytvoriť určitý produkt a v rámci návrhu si môže vybrať z viacerých technologických riešení. V takomto prípade je vhodné pozvať odborníkov z týchto oblastí pričom je najlepšie, ak sú zároveň zamestnancami spoločnosti a požiadať ich o názor, či je daná technológia skutočne vhodná pre špecifikované požiadavky. Iným príkladom môže byť preukázanie určitých technických vlastností produktu v neskorších fázach jeho vývoja.

*Inšpekcia* je druh stretnutia, ktorý je podobný technickým prehliadkam a to hlavne v tom, že sa tiež zaoberá odbornými aspektmi. Sú medzi nimi však určité rozdiely. Tieto dva typy stretnutí sa líšia hlavne svojím rozsahom, pretože kým technické prehliadky sú zamerané zväčša na celý produkt alebo jeho hlavné súčasti, inšpekcie selektívne skúmajú len jeho určitú menšiu časť alebo ho skúmajú ako celok ale z istého pohľadu napríklad iba jeho rozhrania. Z toho vyplýva, že trvajú kratšie, typicky niekoľko hodín. Adresujú priamo autora skúmanej časti a ich výstupom môže byť buď prijatie so žiadnymi alebo minimálnymi úpravami, prijatie s uistením sa o prepracovaní alebo naplánovanie opätovnej inšpekcie.

*Previerka* (čo je výraz voľne preložený z anglického slova walk-through) je menej formálne alebo aj neformálne stretnutie, na ktorom jeden člen vývojového tímu oboznamuje svojich kolegov so svojou prácou. Cieľom tohto typu stretnutia je hlavne hľadať problémy a anomálie, vylepšovať softvérový produkt, zvažovať alternatívne možnosti návrhu a implementácie a overovať súlad s požadovanými štandardami a požiadavkami.

Špeciálnou kategóriu je *audit* čo je vlastne konfrontácia nejakého produktu alebo procesu jeho vývoja so štandardami, odporúčaniami, plánmi alebo inými formálnymi špecifikáciami a postupmi. Vývojový tím by sa ho mal zúčastniť len ako pozorovateľ a nemal by mať žiadnu možnosť ovplyvniť jeho výsledok. Audity môžu byť interné čiže vedené samotnou vývojárskou spoločnosťou alebo externé, ktoré vedie nejaká tretia strana a ktorý sa zväčša vykonávajú za účelom získania certifikátu. Ako príklad môžeme uviesť certifikát bezpečnosti, pretože tá zohráva kľúčovú úlohu napríklad v bankovníctve a kvalita s ňou podľa nášho názoru súvisí.

## Prehliadky zdrojového kódu

Túto techniku uvádzam v eseji kvôli tomu, že ma osobne veľmi zaujala myslím si, že má veľký potenciál priamo zvyšovať kvalitu a to nielen výstupu ale aj zdokonaľovať samotných vývojárov.

Prehliadka zdrojového kódu patrí medzi statické metódy analýzy produktu, aj keď existujú názory, že väčšina programátorov si program simuluje v hlave a tým je to vlastne aj analýza dynamická. Podľa mojich skúseností je však nereálne pochopiť, ako niektoré procesy presne fungujú a to najmä vo väčších projektoch a v praxi sa to uplatňuje len pri určitých menších, zväčša kritických častiach zdrojového textu.

Proces sa môže uskutočniť buď automatizovane s použitím nejakého softvérového prostriedku alebo manuálne.

Automatizovaná prehliadka kontroluje, či sú splnené vopred zadané pravidlá. Tie sa môžu týkať buď určitých konvencií ako napríklad formátovania a pomenovania premenných a metód alebo iných vlastností ako výkon, spoľahlivosť alebo bezpečnosť. Systému je možné napríklad zadať aby kontroloval odchytyvanie výnimiek použitím triedy, ktorá je rodičovská pre všetky ostatné špecifickejšie (najčastejšie označovanej ako *Exception*), čo je veľmi nežiaduca praktika, nakoľko môže znehodnotiť celý systém výnimiek.

Manuálna kontrola, tiež označovaná aj ako čítanie kódu sa najčastejšie vykonáva buď samostatne alebo v pároch, pričom autor zvyčajne predvádza svoj kód spolupracovníkovi. Samostatná kontrola by mala byť podľa môjho názoru pre každého vývojára samozrejmosťou, pretože by mu malo záležať na kvalite jeho odvedenej práce. Spolupráca v pároch má potom podobný charakter, ako previerka.

Rád by som vyzdvihol práve takúto formu kontroly, pretože si myslím, že v mnohých prípadoch môže okrem zvyšovania kvality zvyšovať schopnosti jednotlivých členov tímu pracovať kvalitne a efektívne. V praxi sa mi totiž neraz stalo, že pri pohľade na spolupracovníkov kód som sa naučil robiť mnoho vecí efektívnejšie, vyvarovať sa určitým druhom chýb a dokázal zdokonaľovať svoje vlastné zaužívané postupy. Sú to práve tie znalosti, na ktoré veľmi neexistuje príručka a ktoré sa nedajú systematicky učiť. Na druhej strane som často v cudzom texte zbadal možné chybové stavy alebo vedel zefektívniť niektoré algoritmy a nazdávam sa, že pri pravidelných prehliadkach v pároch by k tomu dochádzalo pomerne často.

Automatizovaná a manuálna kontrola sa najčastejšie kombinujú a to tým spôsobom, že sa najprv automaticky kontrolujú projektovo nešpecifické pravidlá a potom sa prebieha čítanie kódu v kontexte daného projektu.

## Testovanie alebo dynamické metódy

Okrem manažmentu kvality je testovanie ďalším významným procesom, ktorý vplýva na akosť softvérových produktov. Zaoberá sa hlavne dynamickými metódami čiže takými, pri ktorých je nutné softvér spustiť. V starších klasických modeloch napríklad

vo vodopádovom modeli figuroval v rámci vývoja ako samostatná fáza, ktorá nasledovala po implementácii avšak dnešný trend hlavne v súvislosti s agilnými metódami je taký, že sa vyskytuje v celom vývojovom procese a dokonca existujú techniky, ktoré sú na ňom priamo založené.

Testovanie je dynamická verifikácia správania sa programu na konečnej množine testovacích vstupov vhodne vybraných obyčajne z nekonečného množstva možných vstupov oproti očakávanému predpokladu [4].

*Dynamická* znamená, že testovanie vždy implikuje vykonanie programu na daných vstupoch. Pritom treba brať samozrejme do úvahy skutočnosť, že výstupy sú závislé nielen na nich ale aj na jeho vnútornom stave, čo môže byť pri väčších systémoch považované až za nedeterministické.

*Konečná* množina vstupov vyjadruje, že aj pre jednoduchý program by otestovanie programu na celej množine možných vstupov trvalo mesiace až roky. Preto sa táto množina považuje za nekonečnú a jej celý prechod za nemožný. S tým súvisí aj dôležitá skutočnosť, že testovaním je možné zistiť, že systém obsahuje chyby ale nikdy nie to, že je bezchybný.

*Vybrané* stavy a spôsob, akým sa vyberajú je to, čo od seba odlišuje testovacie techniky. To, ako vyberať vhodné stavy vzhľadom na dané podmienky je veľmi zložitý problém, ktorý okrem testovania súvisí aj s analýzou rizík.

Ak sú neznáme *očakávané* predpoklady pri daných vstupoch a stave, nie je možné previesť testovanie, pretože nie je možné rozhodnúť, či bol vrátený správny výstup. Tieto predpoklady sa obyčajne odvodzujú zo špecifikácie alebo pri menej dôležitých prípadoch aj zo skúseností a určitej globálnej predstave o štandardnom sa správaní softvérových produktov.

Testovacie metódy je možné rozdeliť z viacerých hľadísk. Podľa úrovne sa členia na testy súčiastok (unit testy), integračné testovanie a testovanie systému ako celku. Kým testy súčiastok sa snažia chápať každý malý celok kódu (v objektovom modeli je to zvyčajne trieda) ako samostatnú nezávislú jednotku a pre každú z nich definujú samostatné testy, integračné testovanie sa zameriava na ich integráciu a spoluprácu. Najvyšším stupňom je testovanie celého systému, pričom vtedy je chápaný ako čierna skrinka a testuje sa jeho správanie navonok, prípadne spolupráca s inými systémami.

Ďalším kritériom je cieľ, ktorý je testovaním sledovaný. Spomeniem niekoľko druhov, ktoré ma najviac zaujali.

*Regresné* testovanie sa používa pri modifikácii systému, pričom sa opakovane testujú tie isté časti, aby si zistilo, či v ňom zmena nevytvorila nežiaduce a neplánované efekty.

Testovanie *výkonnosti* môže často súvisieť priamo s overovaním používateľských požiadaviek a používa rôzne metriky a metódy porovnávania. V niektorých oblastiach napríklad u hier panuje konkurenčný boj založený v prvom rade na tomto atribúte.

Testovanie *záťaže* sa snaží zaznamenať, ako sa bude systém správať pri rôznej miere zaťaženia, aké je jeho maximálne možné zaťaženie a v akom stave sa bude nachádzať po jeho prekročení.

Menej známym je testovanie typu *back-to-back*, pri ktorom prevádza množina rovnakých testov na dvoch rôznych verziách softvéru a výsledky sa potom porovnávajú.

Test *použitelnosti* určuje, aké je jeho používanie jednoduché pre zákazníka, ako efektívne s ním môžu vykonávať požadované úlohy a ako dobre sa dokáže spamätať z chybových stavov. Pri tomto type testovania sa obyčajne uplatňujú postupy zberu informácií od budúcich používateľov systému buď priamo pozorovaním ich činnosti alebo nepriamo napríklad vo forme dotazníkov.

## Životný cyklus procesu testovania

V súvislosti s moderným prístupom k testovaniu hlavne v rámci agilných metód existuje proces testovania paralelne s vývojovým procesom a rozvíja sa spolu s ním. V rámci jeho životného cyklu možno identifikovať tieto nasledujúce fázy.

*Plánovanie* sa vyskytuje v etape celkového plánovania projektu. V nej sa určuje rozsah testovania, pridelujú sa úlohy a zodpovednosti za jednotlivé časti procesu a zdroje a technické prostriedky na jeho vykonávanie.

*Určenie testovacích prípadov* určí v závislosti od úrovne a techniky testovania aké testovacie prípady sa majú vykonať a špecifikuje pre ne očakávané výsledky.

*Vytvorenie testovacieho prostredia* predstavuje tvorbu a nastavenie prostredia a softvérových nástrojov pre vykonávanie a zaznamenávanie jednotlivých testov.

Pri *vykonávaní* testov by sa mal uplatňovať princíp vedeckého experimentovania a s tým súvisiace prísne dodržiavanie definovaných testovacích procedúr a dokumentovanie všetkých pozorovaných stavov.

*Vyhodnotenie výsledkov testov* upriamuje pozornosť na tie testy, kde systém nezareagoval podľa očakávania. Nie vždy musí ísť nutne o chybu, niekedy môžu byť označené za ruch. Na odstránenie chyby je často potrebné lokalizovať jej príčinu, čo môže byť často veľmi zdĺhavý a namáhavý proces.

*Analýza výsledkov* súvisí viac s procesmi vývoja a testovania, pričom sa z dlhodobých výsledkov usiluje usúdiť, ktoré aspekty vývoja je potrebné posilniť a aký efektívny je testovací proces.

## Záver

Manažment kvality a testovanie sú oblasti, ktoré sa zaoberajú otázkou, ako zvyšovať kvalitu vytváraného softvérového produktu.

Prvá sa zaoberá statickou analýzou a kvalitou samotného procesu vývoja, od ktorého sa potom odvíja aj kvalita výrobku. Pozostáva z viacerých podprocesov, pričom táto esej sa vo väčšej miere venovala rôznym druhom stretnutí a prehliadok a špeciálne technike prehliadky zdrojového kódu, v ktorej osobne vidím veľký potenciál pre zvyšovanie kvality.

Druhá pozostáva hlavne z dynamickej analýzy a dá sa členiť z hľadiska úrovne aj z hľadiska požadovaného cieľa. V moderných metodológiách vzájomne koexistuje s vývojovým procesom, pričom vzniká už pri plánovaní.

V čase, keď kvalita získava stále viac a viac na význame si podľa môjho názoru žiadny vývojársky tím nemôže dovoliť ignorovať tieto oblasti alebo sa im venovať len

nesystematicky. V takom prípade si totiž konkurencia môže vyberať z pestrej ponuky postupov a techník ktoré sú stále sofistikovanejšie a efektívnejšie.

### **Použitá literatúra**

1. Bieliková M.: Softvérové inžinierstvo – princípy a manažment, Vydavateľstvo STU, Bratislava, 2000
2. Nelson S. Schumann J.: What makes code review trustworthy?, *Proceedings of the 37th Hawaii International Conference on System Sciences*, 2004
3. Oh J.S, Choi H.J.: A Reflective Practice of Automated and Manual Code Reviews for a Studio, *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05) - Volume 00 ICIS '05*, IEEE Computer Society Washington DC (2005), 37-42
4. PMI Standards Committee: A Guide to the Project Management Body of Knowledge, Maryland, USA, 2004
5. Wheeler S., Duggins S.: Improving software quality, *Proceedings of the 36th annual Southeast regional conference*, ACM Press, New York(1998), 300-309

### **Annotation**

#### *Software quality management and testing*

Quality management and testing have long tradition in engineering and engineering approach and since its formal formation, they are strongly connected. Software engineering as one of the youngest disciplines has natural demand to integrate similar principles into its processes, which would ensure quality of its outputs considering some facts, which are unique for this domain. This essay provides definition of both areas their main purpose and principles of their work and their division into smaller processes. In quality management it describes different types of project meetings and code-reviews and in testing it deals with classification of test techniques and the life-cycle of testing process.