

# Zabezpečenie kvality softvéru v malej softvérovej firme – áno alebo nie ?

CELESTÍN ČERNÁK

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava*

celestin[.]cernak[zavináč]gmail[.]com

**Abstrakt.** Zabezpečeniu kvality v softvérovom projekte je najmä v poslednej dobe prikladaná čoraz väčšia dôležitosť. Nejedná sa pritom len o samotné vykonávanie testov, dôraz je kladený hlavne na kvalitný a efektívny plán testovania a skvalitňovania produktu počas každej etapy jeho vývoja. Jedným z najpoužívanejších postupov je postup známy ako zabezpečenie kvality softvéru (software quality assurance). Môžeme ho definovať ako systematický proces, ktorý zabezpečí, že vyvíjaný produkt bude po dokončení plne vyhovovať želaným kritériám. Táto práca má za cieľ bližšie priblížiť postup zabezpečenia kvality softvéru v malej softvérovej spoločnosti (menej ako 30 zamestnancov). Bližšie predstavím zabezpečenie kvality v jednotlivých fázach vývoja projektu, ich previazanie na iné používané metódy a postupy a pokúsím sa určiť mieru efektívnosti jeho použitia.

## Úvod – čo je to kvalita ?

Vo všetkých oblastiach výroby je v poslednej dobe prikladaná čoraz väčšia dôležitosť zabezpečeniu kvality. Od čias, kedy tento pojem začal hrať dôležitú úlohu vo vývoji akéhokoľvek produktu, ju vývojári a spoločnosti definovali mnohokrát. V súčasnosti je pojem kvalita definovaný viacerými normami, definícia ISO 9001-00 napríklad hovorí, že kvalita je „miera, ktorou základné charakteristiky spĺňajú definované požiadavky“ [3]. Táto definícia je mimoriadne široká a je vhodné ju bližšie rozobrať.

Pri určovaní kvality je nutné vychádzať z vopred určených požiadaviek. Tieto musia byť jasne a presne definovateľné, čo je často krát problematické najmä pri softvérových produktoch. Preto je potrebné jasne a presne definovať, čo má produkt spĺňať ešte pred začatím jeho vývoja. Zároveň je vhodné určiť kritériá a definovať charakteristiky výrobku, pomocou ktorých budeme jeho kvalitu hodnotiť. V porovnaní s klasickým výrobkom je tento proces pri vývoji softvéru omnoho zložitejší a je potrebné brať do úvahy viac vlastností a kritérií.

Tejto problematike sa venovalo a venuje množstvo štúdií a výskumov a postupne sa vyčlenila samostatná oblasť manažmentu kvality s názvom zabezpečenie kvality softvéru (software quality assurance, ďalej len SQA). SQA a s ňou súvisiace procesy sú definované vo viacerých štandardoch a predstavujú odporúčaný postup, ktorý po aplikovaní poskytuje dostatočnú záruku kvality.

Keďže sa SQA zapája do každej etapy vývoja softvérového projektu, prináša so sebou nevýhodu v podobe réžie nutnej na jej implementáciu a tým pádom aj zvýšenie nákladov na vývoj. Na druhej strane, ak je úspešne aplikovaná, prináša vyššiu kvalitu výsledného projektu a nižšie náklady na jeho následnú údržbu. Z týchto dôvodov je potrebné určiť, či je implementácia SQA v softvérovom projekte pre vývojára výhodná. Určiť pomer medzi vynaloženými nákladmi a získanou pridanou hodnotou je náročné hlavne pre softvérové spoločnosti s menším počtom zamestnancov vyvíjajúce niekoľko projektov, pretože ich existencia na trhu môže byť často závislá na kvalite jediného projektu.

## SQA

Norma IEEE 12027 definuje SQA ako „proces, ktorý poskytuje dostatočnú záruku, že softvérový produkt a všetky procesy počas životného cyklu jeho vývoja budú vyhovovať želaným kritériám a dodržia stanovený plán“ [2].

SQA samotné teda nedokáže *zabezpečiť* kvalitu, snaží sa len *poskytnúť dostatočnú záruku*. Dosahuje to definovaním, naplánovaním a vykonaním presne určených aktivít, ktoré túto záruku poskytujú. Správne určené aktivity dokážu včas identifikovať prípadné komplikácie počas každej etapy vývoja. Dodržiavanie dohodnutých postupov je kontrolované pomocou monitorovania jednotlivých procesov, vyhodnocovania priebežných výsledkov a pomocou auditov. Každá etapa vývoja by mala obsahovať kontrolné body, kde sa kvalita dá overiť porovnaním so zaužívanými štandardami.

### Plán SQA

Aktivity a prostriedky zabezpečujúce, že projekt bude vyvíjaný s maximálnym prihliadnutím na kvalitu, sú opísané v pláne SQA. Odporúčaný obsah tohto plánu je definovaný v norme IEEE 730-2002. Kvalitný a prepracovaný SQA plán predstavuje základný kameň jeho úspešného aplikovania vo vývoji. Sú v ňom presne určené všetky prostriedky, ktoré zabezpečia, že vývoj bude prebiehať v súlade s požiadavkami zákazníka na najvyššej kvalitatívnej úrovni. Preto je potrebné zaistiť, aby riešený problém a výsledná podoba jeho riešenia boli jasne a presne definované.

Plán musí brať do úvahy riadenie, vývoj a aj plán údržby softvéru. Sú v ňom navrhnuté aktivity a úlohy na zabezpečenie kvality spolu s požadovanými zdrojmi, vynaloženými nákladmi a zaradením do jednotlivých etáp projektu. Identifikuje jednotlivé artefakty (dokumenty, konvencie a pod.), ktoré budú použité v riadení projektu spolu so spôsobom ich kontroly. Zároveň určuje prostriedky na evidovanie zistených nedostatkov a možnosti ich nápravy.

Aby bol dosiahnutý objektívny pohľad na kritériá definované zákazníkom, plán SQA a použité procesy by mali pripravovať ľudia nezávislí na tíme vývojárov. Skupina pripravujúca plán môže byť súčasťou tej istej organizácie, v poslednej dobe sa však dáva prednosť aj externým organizáciám, ktoré zabezpečia maximálnu objektivitu.

Po stanovení požiadaviek projektu a dokončení plánu SQA je potrebné tento plán implementovať. V prvom kroku to zahŕňa vyškolenie zodpovedných pracovníkov, či už externých alebo interných. V porovnaní s počtom vývojárov je počet pracovníkov zodpovedných za SQA zvyčajne malý. Často krát pracujú aj na viacerých pozíciách. Dôležitosť ich úlohy zabezpečiť SQA to však v žiadnom prípade nezmenšuje, práve naopak, bez zodpovedných pracovníkov by pravdepodobne stroskotal aj ten najprepracovanejší SQA plán.

### **SQA počas životného cyklu projektu**

V každej etape životného cyklu programu sú definované aktivity, ktoré majú zabezpečiť jej kvalitu. Pri uzatváraní jednej fázy je podrobená SQA analýze a až na základe jej výsledku sa rozhodne o prechode do ďalšej etapy projektu. V nasledujúcom texte sú vybrané najdôležitejšie z týchto aktivít.

#### *Zahájenie projektu*

Úlohou SQA v tejto etape je zabezpečiť, že navrhované plány a procedúry sú vhodné, jasné a kontrolovateľné.

#### *Definovanie požiadaviek*

Počas tejto fázy SQA zabezpečuje, že požiadavky sú úplné a konkrétne rozdelené.

#### *Návrh*

SQA pri vytváraní návrhu zaisťuje použitie len schválených návrhových vzorov. Ďalej zabezpečuje, že v navrhovaných komponentoch sú zahrnuté všetky požiadavky. Na záver posunie schválený návrh do konfiguračného riadenia.

#### *Implementácia*

V etape implementácie zabezpečuje SQA kontrolu naprogramovaného riešenia, konfiguračného riadenia a vytváranej dokumentácie. Zároveň odhaľuje nedostatky a definuje kroky potrebné na ich odstránenie.

#### *Testovanie*

SQA v tejto fáze zaisťuje pripravenosť produktu na otestovanie a zabezpečuje, že všetky testy sú vykonané podľa plánov a všetky zistené nedostatky sú zdokumentované a vyriešené.

#### *Akceptácia*

Záverečnou úlohou SQA v životnom cykle projektu je zabezpečiť vykonanie poslednej kontroly a zaisťiť pripravenosť produktu na odovzdanie zákazníkovi.

## Efektívnosť použitia SQA vo svete

Dôležitosť a efektívnosť aplikovania metód a postupov definovaných v SQA bola skúmaná v rôznych prieskumoch. Jedným z najvýznamnejších je prieskum vykonaný v roku 2003 na množine 104 softvérových projektov z Indie, Japonska, USA a európskych štátov [1]. Autori v ňom skúmali pomer medzi kvalitou projektov, v ktorých počas vývoja boli použité postupy SQA, a projektov, kde boli jednotlivé metódy zastúpené v menšej miere. Tabuľka 1 obsahuje niekoľko vybraných metód spolu s ich percentuálnym zastúpením v skúmaných projektoch. Pre zjednodušenie som vybral len záznamy pre USA a Japonsko.

**Tab. 1.** Vybrané metódy a ich percentuálne zastúpenie.

Použité metódy	U SA	Japo nsko
Špecifikácia architektúry	5 4,8%	70,4 %
Funkčná špecifikácia	7 4,2%	92,6 %
Podrobný dizajn	3 2,3%	85,2 %
Revízia dizajnu	7 7,4%	100 %
Revízia kódu	7 1%	74,1 %
Regresné testovanie	7 1%	96,3 %

Z tabuľky jasne vyplýva, že v Japonsku sa jednotlivé metódy používali vo väčšom množstve projektov ako v USA. Kvalitu výsledného softvéru to ovplyvnilo do značnej miery, výsledky sú zosumarizované v tabuľke 2. Zároveň je v nej pre lepšiu ilustráciu pridané porovnanie z podobného prieskumu vykonaného v roku 1990. Autori sa rozhodli určiť produktivitu a kvalitu projektov na základe dvoch kritérií:

- Počet nových riadkov kódu v človeko-mesiach (riadok č. 1)
- Počet nájdených chýb na 1000 riadkov kódu v intervale 12 mesiacov po doručení zákazníkovi (riadok č. 2)

**Tab. 2.** Produktivita a kvalita jednotlivých projektov.

	US A 1990	US A 2003	Japonsk o 1990	Japonsk o 2003
Nové riadky kódu	245	270	389	469

Nájdené chyby	0,83	0,40	0,20	0,02
---------------	------	------	------	------

Z tabuľky 2 jasne vyplýva dôležitosť použitia metód SQA pri vývoji veľkých softvérových projektov. Pre malé spoločnosti zatiaľ podobný prieskum nebol publikovaný, preto je určiť mieru efektívnosti použitia SQA v tomto prostredí o to náročnejšie.

### Konkrétny príklad použitia SQA

Keďže sa táto práca venuje problematike aplikovania SQA v malej softvérovej firme, ako príklad môžem použiť firmu, v ktorej pracujem už takmer tri roky v oblasti konfiguračného riadenia. Keď som do firmy nastupoval, mala asi 10 zamestnancov a pracovali sme na vývoji troch projektov. Pri takom malom rozsahu samozrejme otázka SQA nebola nijako riešená, ale už vtedy to spôsobovalo nemalé problémy vo forme veľkého množstva objavovaných defektov, nezrovnalostí v dokumentácii a podobne. S prijímaním ďalších zamestnancov a zavedením vybraných metodík vývoja však potreba implementovať nástroj na zabezpečenie kvality rapídne vzrástla. V súčasnosti má spoločnosť asi 25 zamestnancov, pracuje na vývoji viacerých projektov a SQA je použitá hlavne vo fáze testovania každého projektu. Hlbšie zavedenie SQA je očakávaným krokom. Bude však pre spoločnosť ekonomicky výhodné?

Na túto otázku sa pokúsim odpovedať podobne ako pri opise SQA počas životného cyklu projektu. Rozoberiem jeho jednotlivé etapy, používané metodiky a navrhmem možnosti aplikovania SQA spolu s odhadovanými nákladmi.

Z pohľadu projektového manažmentu je použitá metóda manažment hodnoty prínosu (Earned Value Management, ďalej EVM), ktorý je síce mimoriadne vhodný pre plánovanie, nedokáže však poskytnúť obraz o kvalite. Z pohľadu metodiky vývoja používame vybrané časti z procesu Rational Unified Process (ďalej RUP). Tieto procesy už priamo prihliadajú aj na kvalitu, je však potrebné ich správne implementovať.

### Zahájenie projektu

V súčasnosti táto etapa vývoja nie je nijak riadená ani kontrolovaná. Spoločnosť síce má definované šablóny projektovej dokumentácie, tieto sú však veľmi podrobné, a preto ich nie je možné použiť ako všeobecnú šablónu pre novo vznikajúce projekty. Zároveň nie je definovaný presný postup vytvárania dokumentácie ani plánov a nie sú určení pracovníci, ktorí by zabezpečovali ich kontrolu.

Možnosť zlepšenia je pritom veľmi jednoduchá. Pomocou SQA stačí presne definovať potrebné šablóny a určiť ich kontrolórov.

Vynaložené náklady na tieto kroky sú zanedbateľné v porovnaní s prostriedkami, ktoré budú v budúcnosti ušetrené vďaka jednoduchšiemu manažmentu zahájenia projektu.

### **Definovanie požiadaviek**

Čo sa týka definovania požiadaviek, ide o mimoriadne dôležitý krok z pohľadu ďalšieho vývoja projektu. Najväčší problém, s ktorým sa stretávame, spočíva často v samotnom zákazníkovi, ktorý nie je schopný jasne a presne určiť svoje požiadavky. Vývoj softvéru tak často pokračuje aj s nejasnými požiadavkami, čo v neskorších fázach spôsobuje nemalé problémy.

Ak by sme však aplikovali odporúčaný postup v SQA, ktorý hovorí, že na nasledujúcu fázu vývoja sa môžeme presunúť až po skončení súčasnej, dosiahli by sme nepochybne lepšie výsledky. Zadávatelia projektov sú často nevypočítateľní, a práve preto musí byť príprava tejto etapy vykonaná naozaj precízne.

Keďže snahou je dosiahnuť úplné a jasne definované požiadavky, tento proces je časovo náročný a vyžaduje väčšie množstvo prostriedkov. V porovnaní so súčasným stavom tak pravdepodobne dôjde k určitému spomaleniu, prínosom ale budú nižšie náklady súvisiace so zmenou podmienok počas implementácie.

### **Návrh**

Do tejto etapy vývoja sa už vo zvýšenej miere zapája aj EVM. Sú v nej podrobne rozpracované jednotlivé požiadavky a je vypracovaný návrh plánu spolu s previazaním na ostatné moduly a časovou náročnosťou. Tým je priamo zabezpečená aj požiadavka SQA na zahrnutie všetkých požiadaviek.

Zaujímavým momentom z pohľadu aplikovania SQA je posunutie schváleného návrhu do konfiguračného riadenia. To ešte pred samotnou implementáciou môže spraviť úpravy, ktoré by inak bolo nutné vykonať až zmenou počas implementácie, samozrejme za zvýšených nákladov. Túto možnosť by som rád zdôraznil, keďže sa to v spoločnosti stávalo pomerne často a vždy to so sebou prinášalo množstvo komplikácií.

Rovnako ako v predchádzajúcich prípadoch, aj tu krátkodobé zvýšenie réžie na zabezpečenie presunu do konfiguračného riadenia ušetrí prostriedky vynaložené na neskoršie opravy.

### **Implementácia**

Najväčší problém etapy implementácie je nedostatočné vytváranie dokumentácie. Jednou z najelementárnejších oblastí, ktorých sa to týka, je napríklad komentovanie zdrojového kódu. To býva často veľkým nedostatkom a mimoriadne komplikuje úpravy zdrojového kódu v určitom časovom odstupe po jeho vytvorení.

SQA v tejto etape priamo zabezpečuje vytváranie dokumentácie, definuje predpísané šablóny a hlavne kontroluje ich dodržiavanie. Myslím, že netreba zdôrazňovať, aký veľký prínos by to malo v bežnej softvérovej spoločnosti.

V súčinnosti s EVM tu SQA zabezpečuje aj kontrolu dodržiavania termínov. V prípade neefektívnej kontroly to môže vyústiť do značných komplikácií, preto je použitie SQA v tejto fáze mimoriadne dôležité.

## **Testovanie**

V našej spoločnosti je použitie metód SQA najvýraznejšie práve v tejto oblasti. Samostatné testovacie oddelenie zabezpečuje vykonávanie testov ako aj vytváranie testovacej dokumentácie. Použitie prostriedkov RUP pre testovanie zaisťuje podrobné zdokumentovanie nájdených nedostatkov. Neustále prebieha aj kontrola používaných postupov a dokumentov a ich prípadná revízia.

Celý testovací proces u nás priniesol zvýšenie kvality takmer okamžite po jeho zavedení a hoci vyžadoval vyššie počiatočné náklady, je rentabilný aj v krátkodobom meradle.

## **Akceptácia**

Posledná fáza vývoja je v našej spoločnosti prepracovaná a previazaná so SQA. Už na konci etapy testovania sú použité akceptačné testy. Posledná etapa kontroly prebieha priamo u zákazníka. Na jej vykonanie sú presne definované postupy, ktoré zabezpečia, že odovzdaný produkt zodpovedá požiadavkám klientov.

## **SQA a ľudské zdroje**

Aplikovanie postupov SQA so sebou nevyhnutne prináša zmeny v ľudských zdrojoch spoločnosti. Pre menšie firmy je postačujúce, aby SQA zabezpečovali interní pracovníci. Či spoločnosť zamestná na tento účel nového človeka alebo len zmení existujúce pracovné úväzky, záleží od mnohých faktorov.

U nás to napríklad znamenalo vytvorenie nového testovacieho oddelenia. Hoci sú počiatočné náklady na tento krok relatívne vysoké, získaná hodnota ich zvyčajne ďaleko prevyšuje a je na manažmente spoločnosti, ako sa s tým vysporiada. Mať vo firme zamestnanca zaoberajúceho sa iba SQA je pre malé spoločnosti často zbytočný luxus, v takých podmienkach postačuje zapojiť zamestnanca nezávislého na tíme vývojárov.

## **Dokumentácia**

Počas všetkých etáp aplikácie SQA v spoločnosti je nutné podrobne dokumentovať všetky podniknuté kroky. Pomocou nich bude v konečnom dôsledku definovaný celkový plán SQA. Hoci odporúčaný postup hovorí, že najprv treba vytvoriť podrobný plán SQA a potom začať s jeho aplikáciou, v našej spoločnosti sa osvedčilo aj zabezpečovanie kvality a vytváranie dokumentácie za behu. Práve tu sa ukazuje drobná výhoda malých softvérových spoločností v porovnaní s veľkými korporáciami: hoci nemajú všetko naplánované od začiatku, dokážu túto stratu pomerne rýchlo dohnať.

## **Záver**

V tejto práci som sa venoval zabezpečeniu kvality softvéru. Bližšie som určil pojem kvality, predstavil som jednotlivé fázy použitia a vytvorenie plánu SQA a pokúsil sa určiť efektívnosť použitia SQA v malých softvérových spoločnostiach. Keďže o tejto

problematike neexistujú relevantné prieskumy, rozhodol som sa použiť jeden konkrétny príklad možnosti aplikovania SQA v procese vývoja. Rozložením na jednotlivé etapy som opísal možné zlepšenia, ktoré by mohli byť inšpiráciou pre iné spoločnosti používajúce podobný postup vývoja. Po analýze a prehodnotení všetkých aspektov použitia SQA môžem povedať, že procesy SQA prinesú zvýšenie kvality a dlhodobé znižovanie nákladov aj v malých softvérových spoločnostiach. Tieto procesy samozrejme nemusia byť aplikované všetky a presne podľa noriem, ak však budú použité po dôslednom naplánovaní a na správnych miestach, zabezpečia dostatočné zvýšenie kvality. Aj pre malé softvérové spoločnosti je teda otázka aplikovania SQA podľa môjho názoru jednoznačne zodpovedateľná – áno.

### **Použitá literatúra**

1. Cusumano, M., MacCormack, A., Kemerer, C., Grandall, B.(2003). Software development worldwide: the state of the practice, *IEEE Software*, 20(6): 28-34.
2. Feldman, S. 2005. Quality assurance: much more than testing. *Queue* 3, 1 (Feb. 2005), 26-29.
3. TC 176/SC (2005). ISO 9000:2005, Quality management systems -- Fundamentals and vocabulary. International Organization for Standardization.

### **Annotation**

*Software quality assurance in small software companies – yes or no ?*

High importance is being assigned to the quality in software project over the last decades. Apart from simple testing, the importance is put mostly on solid and effective testing plan and improving the product quality during each stage of its development. One of the most used procedures is called the software quality assurance. It can be defined as a systematic process providing that developed product will meet the expected requirements after its completion. The objective of this paper is to explain the procedure of software quality assurance in small software company (with less than 30 employees). I will introduce the quality assurance during product development stages, their connection to other used methods and procedures and I will attempt to define the effectivity of the quality assurance implementation.