

# Manažment kvality pri vytváraní softvéru

PETER CSÓKA

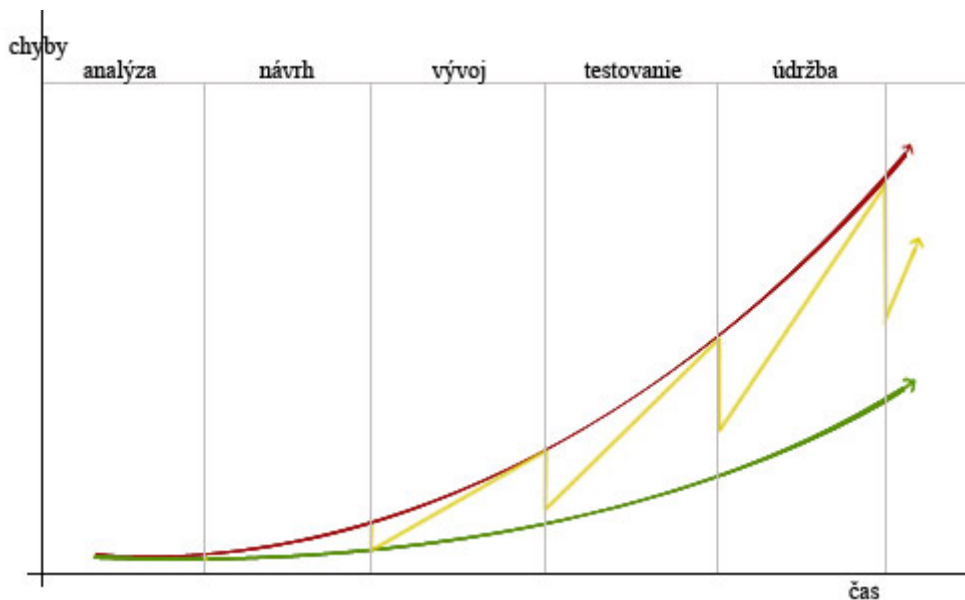
*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
csokapeter[zavináč]gmail[.]com*

**Abstrakt.** Vývoj softvérového systému je rozsiahla úloha. Vytváranie naozaj dobrého softvérového systému vyžaduje zaoberať sa aj otázkami kvality. Zabezpečenie systému kvality musí brať do úvahy nové stratégie, nástroje, metodológie a techniky pri aplikovaní metód životného cyklu softvéru. Tento dokument sa zaoberá faktormi ovplyvňujúcimi efektívne riadenie kvality, úlohou manažmentu a snaží sa poukázať na dôležitosť dodržania štandardov. Ďalej analyzuje rôzne prístupy na dosiahnutie softvérovej kvality a to prostredníctvom hodnotenia kvality, merania a štandardov. Osobitná časť sa venuje porovnaniu zabezpečenia kvality v malých a veľkých tímoch.

## Úvod

Napriek neustálemu rozvoju informačných technológií kvalita softvéru a skvalitnenie procesov vývoja nedostáva dostatočný priestor. Väčšina spoločností si už uvedomila dôležitosť manažmentu kvality, ale bez podporných prostriedkov sú schopné sústrediť sa len na niektoré oblasti a zaoberať sa len niektorými z faktorov ovplyvňujúcimi zvyšovanie kvality.

Najvhodnejším riešením sú metódy pokrývajúce celý životný cyklus softvéru, ktoré vznikli za účelom čím skôr identifikovať chybu, a tým ušetriť na nákladoch spojených s opravou. Nasledujúci obrázok (Obr. 1) ilustruje, ako sa zvyšujú náklady vynaložené na odhalenie chýb v jednotlivých etapách životného cyklu softvéru. Zelená krivka ilustruje prípad, keď sa v určitých intervaloch robí znovuhodnotenie pôvodných plánov, a tým sa náklady v nejakej miere znižujú. Hnedá krivka predstavuje prípad bez návratov k znovuhodnoteniu počas etapy vývoja. Krivka žltá je krivka, ktorou je možné charakterizovať náklady spojené s etapou vývoja skutočného softvérového systému.



**Obr. 1.** Časovo zvyšujúce sa náklady na opravu chýb [1].

Z hľadiska používateľa sa tiež zvyšujú očakávania týkajúce sa kvality softvéru. Každá domácnosť je vybavená počítačom, ľudia používajú služby Internetu, ktoré im uľahčujú prácu, a tým aj šetria čas. Realizujú sa platby cez internet, nákupy, posielajú sa správy, atď. Ľudia používajú softvér, ktorý im pripadá ako najvhodnejší. Nekvalitný softvér, ktorý má napríklad výpadky alebo neprehľadné používateľské prostredie sa dostáva mimo záujmu používateľov, čo spôsobuje nemalé škody softvérovej fabrike.

## Základy manažmentu kvality

Systém manažerstva kvality obsahuje štruktúru organizácie, kompetencie, postupy, procesy a zdroje vyčlenené na riadenie kvality. Je zároveň nástrojom, ktorý umožňuje riadiť a kontrolovať organizáciu z hľadiska kvality, stanoviť politiku a míľniky na dosiahnutie vytýčených dlhodobých cieľov. Tento systém, založený na princípoch prevencie a neustáleho zlepšovania, je dynamický a schopný kedykoľvek sa prispôbiť novým potrebám organizácie. Pravidelné revízie zaručujú jeho pružnosť a efektívnosť a stanovujú spôsob jeho neustálej aktualizácie. Systém manažerstva kvality je systém zameraný na procesy roztriedené do troch skupín [4,6]:

- Strategické procesy sú procesy určené na stanovenie a kontrolu cieľov a politiky organizácie. Tieto procesy sú riadené priamo manažmentom podniku: plánovanie kvality, riadenie zdrojov, preskúvanie systému.

- Kľúčové procesy sú procesy zamerané na činnosti, za účelom poskytovať služby zákazníkom a uspokojovať ich požiadavky a potreby: spracovanie ponuky, plán práce a jej sledovanie, údržba, dokumentácia, rozmnožovanie a odovzdanie.
- Podporné procesy - hoci tieto procesy priamo nesúvisia s činnosťami uskutočňovania politiky, svojimi výsledkami ovplyvňujú úroveň kvality strategických a kľúčových procesov. Patrí sem napríklad údržba infraštruktúry.

### **Úlohy manažmentu kvality**

- Zintenzívniť neustále zlepšovanie procesov, postupov a produktov
- Dodávať produkty a poskytovať služby v zhode s potrebami zákazníka
- Uskutočňovať prieskumy o spokojnosti zákazníka
- Poskytovať zákazníkom najlepšie možnosti za konkurenčné ceny
- Rešpektovať záväzne stanovené dodacie lehoty
- Prispôbiť sa v maximálne možnej miere vývoju technológií na trhu
- Sústavne školiť pracovníkov, motivovať ich a oboznamovať s dôležitosťou riadenia a vývoja systému kvality a zapojiť doň celú organizáciu.
- Zaručiť kontinuitu a rentabilitu podniku vytváraním hodnôt a usilovať sa o čo najlepšie operatívne, ekonomické a finančné výsledky.

Činnosť manažmentu kvality sa preveruje v rámci interných auditov. Interné audity sa vykonávajú za účelom zistenia, ako sa plnia činnosti popísané v dokumentácii systému manažérstva kvality a overenia ich účinnosti.

Interné audity pokrývajú všetky úseky a činnosti systému manažérstva kvality.

Pri auditoch sa zisťujú oblasti zlepšenia, ako aj silné stránky projektového úseku. Venuje sa aj nedostatkom zisteným pri predchádzajúcich auditoch.

### **Pohľady na kvalitný softvér**

Prečo sa zaoberáme kvalitou softvéru? Lebo skúsenosť ukazuje, že softvér môže obsahovať chyby. Prečo softvér obsahuje chyby?

- Ľudia sa pri vykonávaní svojej práce dopustia chýb.
- Aj skúsení programátori majú priemerne jednu chybu v každých 10-tych riadkoch zdrojového kódu.
- Polovica týchto chýb sa opraví pri kompilácií.

- Pri testovaní sa odhalia ďalšie chyby, ale aj tak 15 percent chýb ostane pri odovzdaní softvéru klientovi [2].

Na kvalitu softvéru môžeme pozeráť z rôznych pohľadov. Z hľadiska koncového používateľa sú dôležité vlastnosti ako funkcionálnosť, spoľahlivosť, použiteľnosť alebo prenositeľnosť. Bežného používateľa nezaujíma vnútorná architektúra softvéru, ani spôsob, akým bol vyvíjaný. Pre vývojára je dôležité, aby softvér bol udržiavateľný, modifikovateľný, testovateľný a aby obsahoval čitateľnú dokumentáciu. Úlohou manažéra pri kvalite je, aby zabezpečil vytvorenie systému na stanovený čas a taktiež zabezpečil neprekročenie rozpočtu definovaného na vývoj softvérového produktu a dodržanie zmluvných požiadaviek.

Pojem dobrý softvér viackrát používam v tejto eseji. Práve z tohto dôvodu uvediem v bodoch, čo sa pod týmto slovným spojením myslí: Čo teda znamená dobrý softvér z hľadiska trhu? [2,5]

- Vyhovuje požiadavkám používateľov, robí to, čo od neho používateľ očakáva.
- Je vytvorený na vopred stanovený čas a je lacný.
- Je stabilný.
- Má používateľsky priateľské prostredie a rýchlu obsluhu.
- Jeho inštalácia je bezproblémová.

## Hodnotenie kvality pomocou modelov

Takmer každá organizácia má svoje interné smernice na dodržanie štandardov a monitorovanie kvality. Štandardy pomáhajú pochopiť procesy v rámci daného projektu a uľahčujú aj komunikáciu medzi zainteresovanými členmi. Práve preto musí byť presne definovaný model kvality. Tento model musí odzrkadľovať, že softvérový produkt dodaný danou fabrikou disponuje dvoma základnými vlastnosťami kvality, a to spoľahlivosťou a udržiavateľnosťou.

Charakteristiky, vlastnosti a atribúty kvality tvoria profil kvality. Dobrý softvér je vytvorený podľa daného profilu kvality. Profil kvality vytvorí vývojári spolu so zohľadnením používateľských požiadaviek a potrieb. Takýto profil obsahuje napríklad body na dodržanie termínov a neprekročenie vopred stanovenej ceny alebo požiadavky, aby softvér nevyžiadal inštaláciu cudzích nástrojov na strane klienta. Môže existovať softvér, pri ktorom ide o ľudské životy – v takýchto prípadoch sa musí venovať osobitná pozornosť vlastnostiam ako: spoľahlivosť, kontrolovateľnosť a integrita súčiastok.

Pri vývoji softvéru sú zavedené rôzne smernice, ktoré riadia a organizujú činnosti. Činnosti sú v tomto prípade procesy, z ktorých môžeme vytvoriť model obsahujúci vstupy, výstupy a podmienky vykonávania.

Pri vývoji je potrebné merať atribúty kvality. Na tento účel existujú rôzne modely: ISO 9001-2001, CMM (Capability Maturity Model), SPICE/ISO 15504

(Software Process Improvement and Capability dEtermination), CMMI (Capability Maturity Model Integration).

Modely na základe rôznych kritérií pozorujú a hodnotia spoločnosť a zaradia ju podľa miery vyspelosti. Vyspelosť znamená, v akej miere sú procesy presne definované, riadené, hodnotené a účinné. Niektoré modely hodnotia celú organizáciu, vrátane všetkých procesov (CMM, CMMI), niektorý sa sústreďia len na vybrané procesy (SPICE ISO 15504)[3].

## Meranie softvéru a metriky

Preverenie kvality pomocou modelov je drahé, časovo náročné a zdržuje vytvorenie systému.

Meranie slúži na to, aby bolo možné zo softvéru a procesov vytvoriť numerickú hodnotu. Tieto hodnoty sa porovnávajú s očakávanými a takýmto štýlom sa potom hodnotí softvér. Napríklad spoločnosť zavedie nový nástroj na testovanie softvérov. Ak sa po zavedení nástroja počet detekovaných chýb zvýši, tak softvér je efektívnejší ako jeho predchodca.

Sú dve hlavné oblasti, kde meranie môže byť užitočné [3,5]:

- Odhady na systém – merajú sa komponenty a určí sa pravdepodobnosť výskytu chyby
- Identifikácia chybových komponentov – identifikujú sa komponenty, ktoré porušujú normy

Metrika zahŕňa všetky merania, ktoré nejakým spôsobom súvisia so softvérom, s vnútornými procesmi alebo s dokumentáciou (sem patrí napríklad aj počet riadkov zdrojového kódu vyjadrený ako „rozmer“ produktu) [4]. Napriek tomu, že sa v dnešnej dobe väčšie organizácie zaoberajú metrikami v rámci hodnotenia kvality, meranie nie je až také populárne a rozšírené. Prečo? Môže to byť napríklad z dôvodu, že nie je známe aké výhody prinášajú. Možno, že štúdie a metriky na organizáciu softvérových procesov v rámci spoločnosti sú na tak nízkej úrovni, že nevedia definovať kladné stránky tohto typu hodnotenia kvality, a ani neexistujú žiadne interné dokumenty a štandardy na zber a analýzu údajov takéhoto typu.

## Zabezpečenie kvality v malých tímoch/firmách

Konkurencia je hlavným faktorom, ktorá núti aj malé firmy zaoberať sa otázkami kvality. Problém je, že malé firmy spravidla majú nedostatočne vybudovanú organizačnú štruktúru. Samozrejme sú rozdelené základné roly ako programátor, analytik, dizajnér, ale viac-menej každý člen sa podieľa na vývoji všetkých častí projektu. Majú nedostatočné vedomosti, nevedia predpovedať skutočné efekty plánovaných zmien. Veľmi veľa závisí aj od postoja ľudí aj od toho, aké majú názory.

Od malých tímov nie je možné očakávať z dôvodu ohraničenia finančných a časových zdrojov, aby zaviedli systém podľa známych modelov. Môžu sa však

zaoberať niektorým z modulov a aplikovať ich pri práci. Avšak každý člen musí byť pripravený na zmenu, musí poznať súčasnú situáciu a ciele. Musí vedieť presne určiť procesy, ktoré je potrebné zmeniť, a taktiež vedieť zhodnotiť prínosy uskutočňovaných zmien. Napriek tomu, že sa malé tímy neinšpirujú žiadnym známym modelom, nejaká vnútorná organizácie však predsa musí byť, vzhľadom najmä na to, že vytvárajú systém, ktorý bude reálne nasadený a bude ho používať určitý okruh ľudí. Ako teda zabezpečiť kvalitu? Predovšetkým zdokumentovaním všetkých úloh, testovacích vstupov a výstupov a uvedením minimálnych požiadaviek na inštaláciu softvéru a návodom na používanie.

Doteraz som písal o tom, aké dôležité sú štandardy, modely a tiež aké majú výhody veľké firmy. Ďalšia otázka môže byť teda: Majú nejaké výhody aj malé tímy oproti veľkým, alebo vedľa byť aspoň niekedy lepší?

Výhodou malých tímov/organizácií je predovšetkým lepšia komunikácia. Každý pozná každého, nie je problém priamo dávať námety alebo odporúčania na vedenie. Predstavme si prípad, že programátorovi napadne nejaká vec, ktorá by zefektívnila prácu v tíme. Svoj nápad zdieľa ostatným a ihneď sa môžu rozhodnúť, či ho akceptujú, alebo ho zamietnu. V prípade veľkej organizácie programátor musí nahlásiť vedúcemu svoje nápady, vedúci ďalej manažmentu. Ak príde potvrdenie od manažmentu, tak sa má vytvoriť dokument o analýze problémovej oblasti, o cieľoch vrátane návrhu riešenia. Tento dokument musí manažment schváliť, resp. doplniť chýbajúce časti. Ako vidíme, takýto proces je veľmi zdĺhavý a ani sám si neviem predstaviť takého programátora, ktorý by chodil za manažmentom s vlastnými návrhmi. V súvislosti s tým vznikne aj ďalšia otázka: Koľkých programátorov zaujíma vrcholový manažment a vedenie niektorej obrovskej organizácie, a koľkí by bojovali za niečo, z čoho pravdepodobne nič nezužitkujú. Samozrejme sú zamestnanci, ktorí chcú urobiť kariéru týmto spôsobom, chcel som len naznačiť, že tendencia takýchto prípadov je zanedbateľná u väčších firiem.

Ďalšia výhoda menších tímov je, že vďaka jednoduchšej organizačnej štruktúre sa menšie tímy rýchlejšie vedľa „pustiť sa do práce“. Uvediem príklad: malá softvérová fabrika dostala nový projekt. Vedúci tímu rozdelí prácu medzi členmi a každý robí to svoje. U veľkých firiem toto celé prechádza cez projektových manažérov, ktorí poveria manažment kvality, aby sa začali vytvárať vhodné dokumenty. Ďalej tím zodpovedný za dokumentáciu vytvorí analýzu, technický dizajn, a až potom sa k tomu dostanú vedúci tímov, ktorí rozdelia jednotlivé úlohy v rámci tímu. Samozrejme aj práca programátorov sa má dokumentovať, spolu s testovaním a až potom môže nasledovať odovzdanie a nasadenie softvéru.

Týmto som nechcel povedať, že tieto formality sú zbytočné – bol by to nezmysel najmä z dôvodu, že v celom dokumente rozoberám problematiku a píšem o potrebách dodržania štandardov – len som chcel poukázať aj na nevýhody vychádzajúce z vlastných skúseností. V praxi ja ako programátor v jednej firme môžem potvrdiť, že niekedy vyplňanie dokumentov a čakanie na povel od nadriadených zaberá viac času ako samotná práca na projekte.

## Záver

Táto esej sa snažila zdôrazniť dôležitosť existencie manažmentu kvality a nepodceňovať otázky kvality. Identifikovala vnútorné procesy ovplyvňujúce aspekty vývoja softvéru a predstavila úlohy manažmentu. Ďalej sa zaoberala hodnotením kvality pomocou modelov a metrík a boli naznačené aj výhody a nevýhody prístupov k vývoju softvérového systému pri malých a veľkých tímoch. Čo k tomu dodať ešte na záver? Každý softvér treba navrhnuť tak, aby spĺňal všetky požiadavky používateľov, a pritom dosahoval najlepšie výsledky pri hodnotení kvality pomocou metrík a modelov.

## Použitá literatúra

1. Bieliková, M.: *Softvérové inžinierstvo. Princípy a manažment*. 1. vyd. Bratislava. STU, 2000.
2. Humphrey, W.: *What What if your Life depended on Software?* 2nd SEPG Conference India (Bangalore, India, February 2000).
3. Nasib, S. G.: *Factors Affecting Effective Software Quality Management Revisited*. ACM SIGSOFT Software Engineering Notes, Volume 30, Number 2. March 2005.
4. Pressman, R.S.: *Software Engineering: A Practitioner's approach*. 5th Edition. McGraw Hill. 2001.
5. Sommerville, I.: *Software Engineering*. Addison-Wesley. 7th edition 2004.
6. William, A.W.: *Some Observations on Software Quality*. ACM Southeast Regional Conference. 1999.

## Annotation

### *Quality management in software development*

The development of a software system is a large task. Creating a really good software system requires to address the issues of quality. Ensuring the quality - new strategies, tools, methodologies and techniques for applying methods of the life cycle of software must be taken into account. This essay deals with the factors influencing the effective quality management, task management and highlights the importance of meeting with standards. The article then analyzes different approaches of achieving software quality through the quality assessment, measurements and standards. A special section is devoted to the comparison of quality assurance in small and large teams.