

# Manažment kvality a vplyv na výsledok projektu

PETER LEDŇA

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
Ledna@pobox.com*

**Abstrakt.** Kvalita je bezpochyby najdôležitejšou vlastnosťou každého výrobku. Základnou myšlienkou tejto eseje je pozrieť sa hlbšie na pojem kvalita softvéru, a to v kontexte manažmentu kvality. Príspevok podrobne analyzuje jednotlivé znaky kvality. Diskutuje o tom, ktoré z nich sú dôležité z pohľadu zákazníka a z pohľadu výrobcu softvéru. Rozoberie sa tu pojem chyba a taktiež jednotlivé rozpory medzi požiadavkami zákazníka a požiadavkami výrobcu na softvér. Na záver sa opisuje manažment kvality ako systém riadenia kvality.

## Úvod

O pojme kvalita výrobku sa hovorí, že je to najstaršia ekonomická kategória. Kvalita nie je len vlastnosť výrobku, ale aj ľudí, ktorí tento výrobok vytvárajú a v praxi aj správne používajú. Z hľadiska pojmu „kvalita softvéru“ sa musíme zamyslieť, čo pojem znamená z pohľadu zákazníka a čo z pohľadu vývojára. Je zrejmé, že tieto dva pohľady sa líšia. Kým zákazník požaduje bezchybný softvér za čo najnižšiu cenu, spoločnosti vytvárajúce softvér musia hľadať rozumnú mieru medzi tým, čo požaduje zákazník a svojimi vlastnými potrebami.

Manažment kvality sa používa vo svete systému riadenia kvality v rôznych oblastiach priemyslu, ale i v tvorbe riadenia kvality softvéru. Manažment kvality zvyšuje kvalitu softvéru, a to takým spôsobom, že neustále vylepšuje proces jeho výroby.

## Čo je to kvalita?

Pod pojmom kvalita môžeme rozumieť celkový súhrn vlastností a charakteristík produktu (výrobku, procesu alebo služby), ktoré ovplyvňujú schopnosť uspokojovať stanovené alebo predpokladané požiadavky zákazníka [2]. Kvalita je schopnosť

inherentných vlastností výrobku plniť požadované vlastnosti. Ak sa chceme však zamyslieť nad pojmom „požadované vlastnosti“, uvedomíme si, že je to pojem veľmi obšírny. Na hlbšie pochopenie si zavedieme iné definície pojmu kvalita, ktoré nám objasnia jeho význam z iných uhlov pohľadu. Podľa [2] si zavedieme nasledujúce definície kvality:

- Transcendentné definície kvality výrobku: kvalita výrobku je nedefinovateľná, každý užívateľ má svoj vlastný názor, ktorý je ovplyvnený vlastným individuálnym chápaním.
- Výrobkovo orientované definície kvality: kvalita výrobku závisí od skutočnosti, či má výrobok také vlastnosti, aké mu predpisuje technická dokumentácia.
- Výrobne orientované definície kvality: výrobok je kvalitný vtedy, ak sú pri jeho výrobe dodržané výrobné a technologické postupy uvedené v technickej dokumentácii.
- Hodnotovo orientované definície výrobku: pod kvalitným výrobkom budeme rozumieť výrobok, vyrobený za prijateľnú cenu pre zákazníka a pri výrobných nákladoch prijateľných pre výrobcu.
- Užívateľsky orientované definície kvality: výrobok je kvalitný vtedy, ak má všetky vlastnosti zhodné s požiadavkami zákazníka a požiadavky zákazníka musia byť implementované do návrhu výrobku a technickej dokumentácie.

Ako je vidieť, tak pojem kvalita je komplexný a nemožno ho zhrnúť do jednej myšlienky. Jednotlivé definície kvality nie sú rovnaké a niekedy si aj navzájom odporujú. Požiadavky zákazníka sú často nereálne, pretože sa riadia heslom „chcem všetko za čo možno najnižšiu cenu“. A naopak výrobca (softvéru) chce vytvoriť čo najmenej za čo možno najväčší zisk. Aby sme pochopili hlbšiu podstatu problému, rozdelíme si vlastnosti softvérového produktu na:

- Užitočné vlastnosti: sú také vlastnosti softvéru, ktoré sú funkčne dôležité pre jeho používanie.
- Neúžitočné vlastnosti: sú také, ktoré sú pre používanie daného softvéru nežiaduce alebo škodlivé. Sem patria aj chyby (vady), ktorými sa budeme zaoberať neskôr.

Užitočné vlastnosti softvéru delíme do jednotlivých tried, ktoré sa nazývajú znaky kvality. Vzhľadom na špecifickú problematiku pri tvorbe softvérových produktov sú tieto znaky kvality odlišné od iných druhov tovarov. Hlavné delenie je nasledovné [4]:

- Externé: sú dôležité najmä pre zákazníka a priamo ovplyvňujú cenu softvéru na trhu.
- Interné: sú dôležité pre výrobcu softvéru a majú priamy dopad na rozvoj schopností vývojového tímu splniť projektové externé požiadavky kvality, vývoj výdavkových plánov a rozvrhu.

**Externé znaky kvality softvéru**

Niektoré z týchto znakov sú si podobné, ale všetky majú úplne iný zmysel. Externé znaky kvality softvéru sú [4]:

- Výdavky: peniaze, ktoré použil zákazník, aby vyvinul, kúpil, použil a udržiaval softvérový systém. Často sa tiež nazývajú celkové výdavky na vlastníctvo.
- Dodržanie termínu: miera, do akej bol dodržaný konečný plán dodania alebo miera, na základe ktorej je softvérový systém dodaný v rámci vyžadovaného časového rámca zákazníkovi.
- Úplnosť: miera, na základe ktorej systém implementuje svoj plánovaný rozsah s dôrazom na špecifické požiadavky a dodacie podmienky.
- Správnosť: miera, do akej výrobok spĺňa špecifikáciu.
- Spoľahlivosť: správanie sa výrobku pri výpadku. Výrobok by nemal pri výpadku systému spôsobiť ani fyzické ani ekonomické škody.
- Robustnosť: schopnosť, s akou môže softvér pokračovať v prípade neplatného vstupu alebo neočakávaného problému.
- Použiteľnosť: úsilie, ktoré treba vynaložiť na to, aby sa dal výrobok používať.
- Efektívnosť: splnenie kritérií na využitie zdrojov počítačového systému, na čas potrebný na realizáciu a ďalších kritérií spojených so samotným vývojom výrobku.
- Výkonnosť: rýchlosť alebo priechodnosť, minimalizovanie času medzi udalosťami vstupu a výstupu systémom, maximalizovanie množstva potrebnej vykonanej práce v danom časovom období.
- Kapacita: schopnosť alebo vhodnosť pre podržanie alebo ukladanie dát alebo informácií. Kapacita môže byť obmedzená návrhom alebo požiadavkami a taktiež operačným systémom.
- Merateľnosť: miera, do akej nie je kapacita, efektivita a výkonnosť softvérového systému limitovaná svojim dizajnom, implementáciou a hardvérom, na ktorom sa vykonáva.
- Kompatibilita: miera, do akej bude softvérový systém fungovať a komunikovať správne, spoľahlivo a robustne s inými podobnými systémami, s ktorými zdieľa rovnaké typy dát, súborové systémy a užívateľské rozhrania. Spätočná kompatibilita sa týka najmä schopnosti softvérových systémov pracovať s predchádzajúcimi verziami, z ktorých bola odvodená.
- Interoperabilita: úsilie, ktoré treba na zabezpečenie spolupráce systému s inými systémami.

- Integrita/bezpečnosť: úsilie, ktoré treba vynaložiť na to, aby bol systém bezpečný.
- Prispôsobivosť: miera, do akej sa systém môže použiť bez modifikácie v iných aplikáciách alebo prostrediach, pre ktoré bol vytvorený.
- Konfigurovateľnosť: miera, do akej môže užívateľ zmeniť operačné parametre systému.
- Presnosť: miera, do akej je vybudovaný systém bezchybný, hlavne čo sa týka kvantitatívnych výstupov. Presnosť sa odlišuje od správnosti; kým presnosť je určenie ako dobre systém pracuje na úlohe, pre ktorú bol navrhnutý, správnosť vyjadruje, či bol systém správne implementovaný.
- Opakovateľnosť: miera, do akej bude systém opakovane produkovať tie isté výstupy, za predpokladu zhodnej množiny vstupov a zhodného operačného systému.
- Budovateľnosť: ľahkosť, s ktorou môže byť softvérový produkt spoľahlivo vybudovaný zo samostatných komponentov.

### **Interné znaky kvality softvéru**

Interné znaky kvality softvéru sú [4]:

- Udržovateľnosť: úsilie, ktoré treba vynaložiť na ďalší vývoj a údržbu výrobku podľa meniacich sa potrieb zákazníka a aj meniaceho sa okolia.
- Pružnosť: úsilie, ktoré treba na modifikáciu výrobku v prevádzke.
- Prenosnosť: úsilie, ktoré treba na prenos výrobku z jednej platformy na inú.
- Znovupoužitelnosť: miera, do akej možno jednotlivé časti výrobku znovu použiť pri iných podobných aplikáciách.
- Čitateľnosť: ľahkosť, s akou môže vývojár čítať alebo porozumieť zdrojovému kódu alebo technickej dokumentácii systému.
- Testovateľnosť: úsilie, ktoré treba vynaložiť na testovanie vlastností výrobku, napr. či vykazuje požadované správanie.
- Zrozumiteľnosť: ľahkosť, s akou môže niekto pochopiť softvérovému systému na systémovo-organizačnej a tiež na detailnej úrovni. Zrozumiteľnosť v porovnaní s čitateľnosťou súvisí so zviazanosťou a súdržnosťou systému vo všeobecnejšej rovine. Zahŕňa porozumenie nielen toho, čo systém robí, ale prečo to robí.

## Chyby a zdroje chýb

Ako už bolo spomínané, chyby patria medzi neúčinné vlastnosti softvérových produktov. S chybami v softvéri sa stretol asi každý človek, ktorý denne pracuje s počítačom. Nie je zriedkavosťou, že aplikácia alebo operačný systém „padne“, pričom to býva v tej najnevhodnejšej chvíli.

Vývoj softvéru prešiel za uplynulé desaťročia mnohými významnými zmenami. Asi najväčší skok z pohľadu redukcie chýb bol zaznamenaný s príchodom tzv. typových jazykov ako je C++, Smalltalk a neskôr Java. Taktiež sa zlepšili aj testovacie či odlaďovacie nástroje. Napriek tomu sa však zdá, že chýb v softvéri neubúda. Čím to je?

Jednou z odpovedí na túto otázku je, že za posledné roky sa zvýšila aj zložitosť softvéru. Tieto chyby, ktoré vyplývajú zo zložitosti vytváraného systému, sú čoraz „zákernejšie“ a ťažšie rozpoznateľné. Zložitosť je zapríčinená najmä užitočnými (ktoré sú potrebné na správne fungovanie systému), ale často aj postrádateľnými (také, ktoré vznikli z „rozmaru“ zákazníka) funkciami systému.

Ďalším významným faktorom, ktorý vplyva na počet chýb je, že zložité systémy nevytvára jeden človek, ale tím ľudí. Tímová práca vnáša do tvorby softvéru chyby, ktoré vyplývajú najmä zo zlej komunikácie medzi jednotlivými členmi tímu. Na veľkých projektoch sa často rozdeľujú jednotlivé úlohy medzi rôzne tímy, pretože nie je možné, aby jedna malá skupinka ľudí daný projekt vytvorila v určenom časovom rozpätí. Pri takýchto veľkých projektoch nastáva podobný problém, a to najmä v zlej komunikácii medzi jednotlivými tímami.

Jeden z najvýznamnejších zdrojov chýb spočíva v zlej, resp. nedostatočnej komunikácii so zákazníkom. Komunikácia so zákazníkom je veľmi dôležitá, pretože si musíme uvedomiť, že daný produkt sa vytvára práve pre zákazníka. Je nepríjemné zistiť pri odovzdávaní produktu, že zákazník takto vytvorený produkt nechcel. V [2] sa uvádza, že až 15% zo všetkých chýb je spôsobených zlou komunikáciou so zákazníkom. Vzhľadom na túto skutočnosť, by sa mal každý výrobca softvéru riadiť heslom „Náš zákazník, náš pán“.

Špeciálnym druhom chýb sú tzv. bezpečnostné chyby. Tieto chyby sa prejavujú až pôsobením tretej strany. Nejde teda o chybu v pravom slova zmysle. Na druhú stranu je však v dnešnej dobe otázka bezpečnosti veľmi dôležitá, pretože žiaden zákazník asi nechce, aby prišiel o dáta, alebo aby sa jeho citlivé údaje dostali do nepovolaných rúk.

## Čo je problematické pre softvérové systémy?

Pri stanovovaní priorit pri tvorbe softvérového systému narážame často na problémy. Tie súvisia s rozdielnymi požiadavkami na kvalitu softvéru z pohľadu zákazníka a z pohľadu tvorca, resp. údržbára systému.

Zákazník často požaduje, aby mal systém čo najviac funkcií, vysokú mieru konfigurovateľnosti a prispôbitelnosti. Z pohľadu vývojára takáto „zbytočná“ funkcionálna zvyšuje zložitosť kódu, čo následne prináša vysoké riziko vzniku chýb. Taktiež, čím je zložitejšia funkcionálna systém, tým sa znižuje miera testovateľnosti, pretože je nutné vytvoriť väčšie a komplexnejšie vzorky testovacích vstupov.

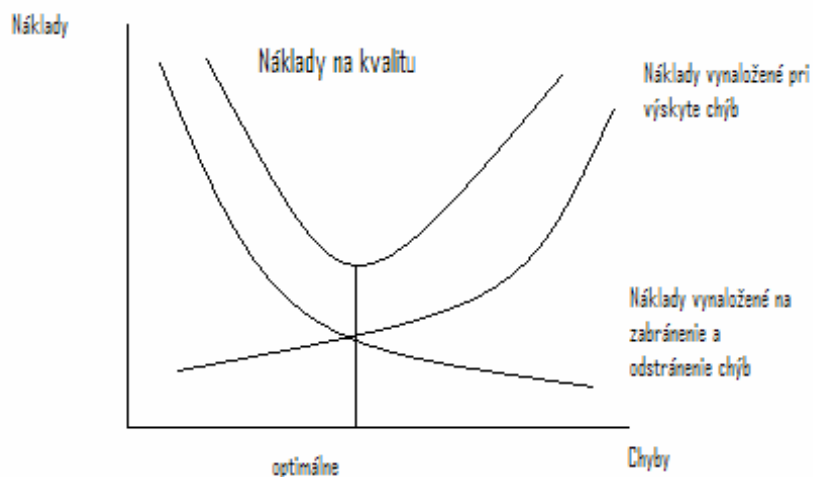
Z pohľadu údržbára sa táto zložitosť prejaví ako zhoršená miera udržiavateľnosti systému.

Ako ďalší príklad môžeme uviesť, že zákazník často požaduje vysokú mieru efektívnosti systému. Z pohľadu vývojára to znamená, že systém je nutné prispôsobiť čo najviac požiadavkám zákazníka a následne upraviť triedy, knižnice, moduly alebo celé podsystémy, ktoré boli použité a prebrané z iných podobných systémov. To zapríčini nielen nižšiu schopnosť znovupoužiteľnosti (či už existujúcich komponentov alebo komponentov, ktoré sa práve v systéme vyrábajú), ale opäť sa tu vynára vysoké riziko vzniku chýb v už odladených komponentoch.

Zákazníci často požadujú, aby vytváraný systém dokázal spolupracovať s inými už existujúcimi systémami, t.j. požadujú vysokú mieru interoperability. Pre vývojára to ale znamená vytvorenie nových rozhraní a taktiež s tým súvisiace ošetrovanie neočakávaných vstupov od iných systémov, t.j. zvýšenie miery robustnosti systému, čo opäť zapríčini zvýšenie pravdepodobnosti vzniku chýb.

Jeden z najzávažnejších problémov však nastáva, keď zákazník nie je schopný dodať úplnú špecifikáciu vytváraného projektu (čo býva prakticky vždy). Špecifikácia je častokrát neúplná a čo je horšie, nekonzistentná. Zákazník často mení svoju špecifikáciu počas vývoja softvéru, čo je už v rozbehnutom vývoji softvéru veľmi nepríjemné a celý projekt sa tým môže predražiť.

Asi najdôležitejšou požiadavkou zákazníka je, aby bol daný softvérový produkt bezchybný. Takáto požiadavka je však prakticky nerealizovateľná, pretože by to trvalo „nekonečne“ dlho a stálo by to „nekonečne“ veľa peňazí. Preto je nutné hľadať optimálne riešenie, t.j. minimalizovať sumu nákladov vynaložených na odstraňovanie chýb. Súvis medzi týmito hodnotami je na obrázku **Obr. 1**.



**Obr. 1.** Súvis medzi nákladmi, chybami a kvalitou

## Manažment kvality

Systém manažérstva kvality má svoje korene v 50-tych rokoch povojnového Japonska. Za jeho zakladateľa možno považovať W. Edwarda Deminga, ktorého postupy na riadenie kvality boli úspešne aplikované počas obnovy zničeného japonského priemyslu. Počas osemdesiatych rokov boli vydané prvé certifikáty ISO 9000. Dnes je manažment kvality široko používaný v rôznych odvetviach priemyslu, a to najmä v automobilovom a elektronickom.

Pod pojmom manažérstvo kvality sa bude rozumieť koordinačná činnosť všetkých oddelení podniku tak, aby vyrábané výrobky mali vlastnosti zhodné s požiadavkami zákazníka [2].

Použitie manažérstva kvality v softvérovom inžinierstve má určité špecifiká. Podstatný rozdiel medzi tvorbou softvéru a napr. výrobou ložísk je v tom, že tvorba softvéru nie je vo všeobecnosti opakovateľný proces. Inými slovami povedané, pokiaľ vytvárame dva rôzne softvérové produkty, potom proces ich výroby nie je rovnaký. Dôvodom neopakovateľnosti procesu výroby je najmä skutočnosť, že každý (zákaznícky) vytváraný softvér je nejakým spôsobom špecifický a pri jeho realizácii nie je možné opakovať tie isté úkony, ako tomu bolo pri iných.

Ďalším dôležitým aspektom odlišnosti medzi softvérom a inými výrobkami je istá obtiažnosť pri meraní jednotlivých znakov kvality. Kým pri už spomínanom ložisku je pomerne jednoduché zistiť meraním jeho priemeru alebo váhy, či vyhovuje alebo nie, pri stanovovaní znakov kvality softvéru to nie je až také jednoduché.

Manažment kvality vychádza z predpokladu, že zlepšením kvality procesu výroby sa zvýši aj kvalita vyrábaného výrobku. Takýto postup sa používa najmä preto, lebo v softvérovom inžinierstve nie je ľahké merať výstupnú kvalitu pomocou nejakej výstupnej kontroly [3]. Zlepšovaním procesu výroby sa znižujú aj náklady na kvalitu, t.j. pri rovnakej cene za vyššiu kvalitu alebo za rovnakú kvalitu zaplatíme nižšiu cenu.

### Aktivity vykonávané pri manažmente kvality

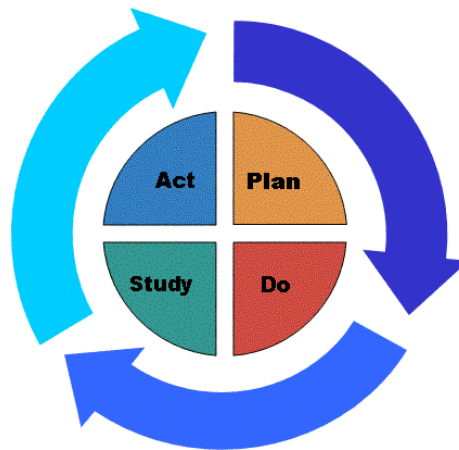
Ako už bolo spomínané, hlavnou úlohou manažérstva kvality je zaistenie kvality pri výrobe softvérových produktov. Hlavné aktivity manažmentu kvality sú [3]:

- Plánovanie kvality: Výber vhodných procedúr a štandardov pre špecifický projekt a ich následné prispôbenie. Vytvára sa tzv. plán kvality, v ktorom sa jasne stanovuje, ktoré atribúty kvality sú najdôležitejšie pre daný vyvíjaný výrobok. Definuje sa proces zabezpečenia kvality (procedúry zabezpečenia kvality sa dokumentujú pomocou tzv. príručky kvality, tieto treba vytvárať na základe postupov, ktoré sa skutočne používajú). Definujú sa metriky, ktoré slúžia na hodnotenie stupňa dosiahnutej kvality.
- Zabezpečovanie kvality: Koordinácia, usmerňovanie vykonávania všetkých tých plánovaných a systematických činností, ktoré treba na to, aby sa vytvorila dostatočná dôvera, že výrobok alebo služba bude spĺňať určité požiadavky. Kvalitu musia zabezpečovať všetci účastníci projektu.

- Riadenie kvality: Sledovanie výsledkov a identifikácia spôsobov eliminácie nevyhovujúcej kvality. Riadenie sa uskutočňuje na základe analýz meraní. Zabezpečenie riadenia často vykonáva skupina na zabezpečenie kvality (SQA group).

Na celý tento postup sa možno pozrieť aj z alternatívneho hľadiska, a to z pohľadu Demingovho PDSA cyklu. Ako možno vidieť z obrázku **Obr. 2**, Demingov cyklus sa skladá zo štyroch častí, ktoré sa „donekonečna“ opakujú v danom poradí. Význam jednotlivých častí cyklu je nasledovný [2]:

- Plánovanie (Plan): Zaznamenanie stavu procesov, analýza štatistických dát a procesov, určenie vplyvových veličín a navrhnutie vylepšení procesov.
- Realizácia (Do): Vyvodenie opatrení z navrhnutých zmien pri fáze plánovania a ich následné zavedenie a presadenie. Následne dostávame zmenené výsledky.
- Preskúmanie (Study): Porovnanie dokumentovaných výsledkov s východiskovým stavom. Ak nastalo zlepšenie, fáza plánovania správne identifikovala vplyv správnych vplyvových veličín, ak nastalo zhoršenie výsledkov, tak identifikácia bolo nesprávna.
- Konanie (Act): celý cyklus musí prebehnúť ešte raz, aby sa vykonala náprava, alebo aby sa dosiahlo ďalšie zlepšenie.



**Obr. 2.** Znáozornenie PDSA Demingovho cyklu.

### Ako merať kvalitu softvéru?

Meranie softvéru je odvodením numerickej hodnoty pre každý atribút softvérového produktu alebo procesu. Softvérová metrika je ľubovoľný typ merania súvisiaci



so softvérovým systémom, procesom alebo dokumentáciou [1]. Metrika vlastne umožňuje softvéru a softvérovému procesu byť kvantifikovateľný.

Ak sa teraz zamyslíme nad tým, ako merať jednotlivé znaky kvality, zistíme, že externé znaky kvality softvéru sa merajú ľahšie ako interné. Je to zapríčinené tým, že externé znaky sa často prejavujú na samotnom produkte a sú tým pádom dobre a objektívne merateľné. Pre interné znaky je situácia o niečo komplikovanejšia, pretože ich hodnotenie je veľmi subjektívne, a preto takéto meranie nie je možné pokladať za absolútne. Ako príklad interných znakov kvality môžeme uviesť zrozumiteľnosť, kde rýchlosť pochopenia systému bude závisieť od osobných schopností a skúseností programátora. Ako príklad externých znakov môžeme uviesť jeden z najdôležitejších znakov kvality softvéru a to spoľahlivosť, ktorá je charakterizovaná strednou dobou medzi poruchami (výpadkami) MTBF a strednou dobou opravy MTTR. Pre spoľahlivosť potom platí [2]:

$$\text{Spoľahlivosť} = \frac{MTBF}{MTBF + MTTR}$$

Meranie kvality softvéru predstavuje, resp. meranie v softvérovom projekte je široká oblasť sama o sebe. Cieľom tejto časti eseje nebolo objasniť spôsob merania v softvérovom projekte, ale poukázať na rozdiely v meraní interných a externých znakov kvality.

### Využitie manažmentu kvality v malých projektoch

Sústava ISO 9000 predstavuje dobrú skupinu noriem na zabezpečovanie kvality. Bohužiaľ, sú s ňou spojené obrovské náklady na dodatočnú pracovnú silu, dokumentáciu a pod. Pre malé projekty (cca 5 ľudí) si je vhodné z nej vybrať v obmedzenej miere niektoré postupy a normy. Jedným zo zaujímavých postupov je statická previerka kódu, kde sa zistí, či sa programátor drží dokumentácie, alebo sú rozpory medzi implementovaným kódom. Taktiež je vhodné vytvoriť si nejakú metriku na identifikáciu kritických častí kódu a túto metriku postupne zlepšovať a spresňovať.

Odporúčal by som aj založenie dokumentácie manažmentu kvality, kde budú zhrnuté pravidlá pre členov tímu a výsledky previerok a testovania zdrojového kódu a dokumentácie.

Málo projekty sú svojim spôsobom špecifické a nemá v nich veľkú cenu dbať na rozsiahlu dokumentačnú časť riadenia kvality, ako to požaduje ISO 9000. Skôr by bolo vhodné zaviesť akúsi „kultúru kvality“, kde každý člen tímu bude niest osobnú zodpovednosť za kvalitu.

### Záver

Kvalita je neoddeliteľnou súčasťou nášho vnímania a posudzovania výrobkov ako takých. Výrobcovia softvéru by si mali uvedomiť, že kvalita je v skutočnosti to, čo sa predáva, resp. to, čo každý zákazník kupuje. Chybám počas vývoja softvéru sa prakticky nedá vyhnúť. Manažment kvality zvyšuje kvalitu softvéru, a to takým

spôsobom, že neustále vylepšuje proces jeho výroby. Ak by sa proces výroby priebežne nezlepšoval, potom by sa začal zhoršovať. Manažment kvality úzko súvisí aj s manažmentom rizík, pretože predpokladá, že kvalitným procesom výroby softvéru sa dá vyhnúť aj prípadným veľkým škodám pri zmene špecifikácie počas vývoja softvéru.

### **Použitá literatúra**

1. Ian Sommerville: Software Engineering 7th edition, Chapter 27, 2004.
2. Ladislav Hulényi.: Poznámky z predmetu Manažment kvality,(2004)
3. Mária Bieliková.: *Softvérové inžinierstvo – princípy a manažment*, Bratislava, 2000.
4. Richard G. Russell.: *Defining Software Quality*, 2004

### **Annotation**

#### *Quality management and its influence on project result*

The quality is with no doubt the most important part of the each product. The basic idea of this essay is to look more deeply into a concept of system quality, in particular, in the context of management of quality. The paper analyses the individual parts of quality characters. It discusses about which of them are important from the customer's view and from the view of the producer of the software. It examines the concept of an error and also contemplates about individual contradictions between the requirements of the customer and that of the software producer. In conclusion it describes the management of the quality as the system of quality control.