

# Manažment kvality a vplyv na výsledok projektu

ONDREJ HLUCHÝ

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
hluchy99@student.fiit.stuba.sk*

**Abstrakt.** Vývoj dobrého softvérového systému je veľmi komplexná úloha. Ak chceme vytvoriť dobrý softvérový produkt, musíme prihliadať na niekoľko metrik kvality softvéru. Zložitosť systému hrá významnú úlohu v procese kontroly a riadenia kvality, pretože vo všeobecnosti vplýva na metriky ako spoľahlivosť, testovateľnosť a udržovateľnosť softvéru. Preto zabezpečenie kvality softvéru treba vziať do úvahy počas celého životného cyklu vývoja softvéru s ohľadom na nové stratégie, nástroje, metodológie a techniky.

Táto esej je prehľadovou štúdiou, ktorej cieľom je poskytnúť ucelený pohľad na manažment kvality v procese vývoja softvéru. Autor v nej odpovedá na otázky, čo to vlastne je manažment kvality a prečo je dôležitý. Ďalej porovnáva existujúce modely a štandardy na zabezpečenie kvality. V závere opisuje vplyv kvality na výsledok projektu na základe vlastných skúseností z praxe.

## Úvod

Náklady na neúspešné IT projekty v USA boli nedávno odhadnuté na 84 miliárd dolárov len za jediný rok [1], takže na kvalite softvéru záleží v dnešnej dobe viac ako kedykoľvek predtým. A záleží na nej nám všetkým, pretože my používame ten softvér. Naša závislosť na softvéri sa zvyšuje z roka na rok či si to uvedomujeme alebo nie. Každým dňom viacej z nás používa softvér na viacero úloh v oblasti informačných technológií (IT), informačných systémov (IS), vo vnorených systémoch v spotrebných tovaroch ako sú napríklad mobilné telefóny a samozrejme v Internete. Riziká spojené so zlyhaním softvéru sa zvyšujú s používaním softvéru. To zahŕňa väčšie riziká pre organizácie, ak softvér zlyhá alebo je nevyhovujúci, a väčšie sklamanie alebo ujmu jednotlivcov, ak softvér nesplní ich očakávania.

Medzitým sa v posledných rokoch zvýšili tlaky na moderné organizácie vrátane firiem. Tieto tlaky – znižovanie nákladov, maximalizácia zisku, zvýšené očakávania zo strany zákazníkov, globálne komunikácie, neustále zmeny a potreba hľadať nové trhy –

sa stávajú tlakmi na softvérové tímy, aby produkovali viac softvérových produktov, rýchlejšie, so zvýšenými očakávaniami na funkcionálnosť softvéru.

Prečo nás IT tak často sklamá? Prečo sa softvér nevytvára bez chýb? Jedna príčina je celkom jednoduchá: IT systémy sú vytvárané ľuďmi a ľudia robia chyby. Toto platí pre každú ľudskú aktivitu, ale v IT máme množstvo okolností, ktoré k chybám napomáhajú:

- IT systémy často vytvárajú tímy ľudí, ktorí nebudú vytváraný systém používať. Za týchto okolností slabá komunikácia medzi tímami a budúcimi používateľmi zvyšuje šance na chyby.
- IT oddelenia sú často neschopné prispôsobiť vytváraný softvér kultúre, procesom alebo cieľom v oblasti, pre ktorú je softvér určený.
- IT systémy sú zložité a ich zložitosť sa stále zväčšuje. Tiež interkomunikácia s inými systémami je stále zložitejšia.

Iba posledná okolnosť je technický problém. Ostatné vymenované okolnosti sa týkajú ľudí, ich schopnosti komunikovať a navzájom si porozumieť a schopnosti učiť sa od druhých a z vlastných skúseností.

## Definície kvality softvéru

Podľa orientácie môžeme vyčleniť štyri rozdielne definície kvality. Tieto sú *produktovo-orientovaná*, *vývojovo-orientovaná*, *používateľsky-orientovaná* a *hodnotovo-orientovaná* [1].

Dve definície kvality podporované IT ľuďmi sú produktovo-orientovaná a vývojovo-orientovaná definícia. V projektoch dávame prednosť definíciám, ktoré nám umožnia merať pokrok a úspech. Chceme fixovať kvalitu na niečo, čo je dodateľné a merateľné.

- V definícii orientovanej na produkt je kvalita založená na dobre definovanej sade metrík, ktoré treba merať objektívnym a kvantitatívnym spôsobom. Môžeme z nich odvodiť akceptačné kritériá, aby sme vedeli objektívne posúdiť kvalitu dodaného produktu. *Príklad: Štandardy ako napríklad ISO 9126 definujú atribúty spoľahlivosť, použiteľnosť, bezpečnosť a funkcionálnosť spolu s metrikami pre ne. "Softvér je spoľahlivý na 98% ak beží nepretržite viac ako 7 dní. Čas na zotavenie je menší ako 1 minúta po každom zlyhaní."*
- Definícia orientovaná na vývoj sa sústreďuje na vývoj softvérových produktov, to znamená ich špecifikáciu, návrh a implementáciu. Kvalita závisí od rozsahu implementovaných požiadaviek. Úspech sa meria ako schopnosť sledovať proces a dodať produkt s ohľadom na dohodnutú špecifikáciu. Produkt budeme verifikovať (je systém správny podľa špecifikácie?), ale ak nezoberieme do úvahy definíciu kvality orientovanú na používateľa (viď nižšie), môžeme zabudnúť validovať (je toto správna špecifikácia?). *Príklad:*

*Opakovateľný, kontrolovateľný proces, ktorý vyhovuje špecifikácii. "Softvér bol vytvorený podľa špecifikácie a obsahuje malé množstvo chýb."*

Existujú dve ďalšie definície kvality, ktoré odrážajú pohľady používateľa a zákazníka. Pretože tlaky na organizácie sa časom menia, to čo predstavuje "kvalitu", sa môže tiež časom meniť. Niekedy sú zmeny taktické – "Musíme znížiť náklady v tomto štvrtroku!" – a niekedy môžu byť strategické – "Chceme byť lídrami na trhu!" – a tieto si môžu protirečiť.

- Používateľsky orientovaná definícia hovorí, že kvalita vyjadruje vhodnosť použitia. Kvalita softvéru by mala byť stanovená používateľom produktu v špecifickej situácii. Rozličné obchodné charakteristiky vyžadujú rozličné typy softvérových produktov; nielen na robenie iných vecí, ale tiež treba dbať na to, že rozliční ľudia môžu vykonávať svoje úlohy odlišne. Táto skutočnosť môže byť subjektívna a nedá sa určiť len na základe kvantitatívnych metrik. Práve používateľsky orientovaná definícia nás nabáda k validácii ako aj verifikácii systému. *Príklad: Vhodnosť pre cieľ. "Môžem robiť svoju prácu efektívne ak použijem tento softvér."*
- Hodnotovo-orientovaná definícia je zameraná na veci, ktoré majú vplyv na firmu ako celok. Kvalita softvéru by mala byť vždy určovaná prostredníctvom úsilia a cenových aspektov. To sa deje komunikáciou všetkých zúčastnených strán: sponzori, zákazníci, vývojári a výrobcovia. *Príklad: Návrat investícií. "Ak vydáme softvér teraz, budeme musieť vynaložiť \$250000 navyše na podporu v prvom mesiaci. Ak sa oneskoríme o mesiac, bude to stáť organizáciu \$1000000 na pokutách a stratu biznisu. Mali by sme softvér vydať alebo ďalej testovať?"*

V jednom projekte môžeme brať do úvahy niekoľko definícií kvality a nemusia si to ani uvedomiť všetci ľudia v projekte. Je dôležité uvedomiť si, že neexistuje žiadna správna definícia kvality. To ako zadefinujeme kvalitu pre určitý produkt, službu alebo projekt závisí od situácie.

## **Fundamentálne koncepcie dokonalosti**

Tieto koncepcie [1] sa používajú v množstve modelov vrátane EFQM (European Foundation for Quality Management) Modelu dokonalosti, ktorý sa používa vo väčšine krajín Európskej Únie. Podobné organizačné modely založené na týchto koncepciách boli vyvinuté aj v iných krajinách.

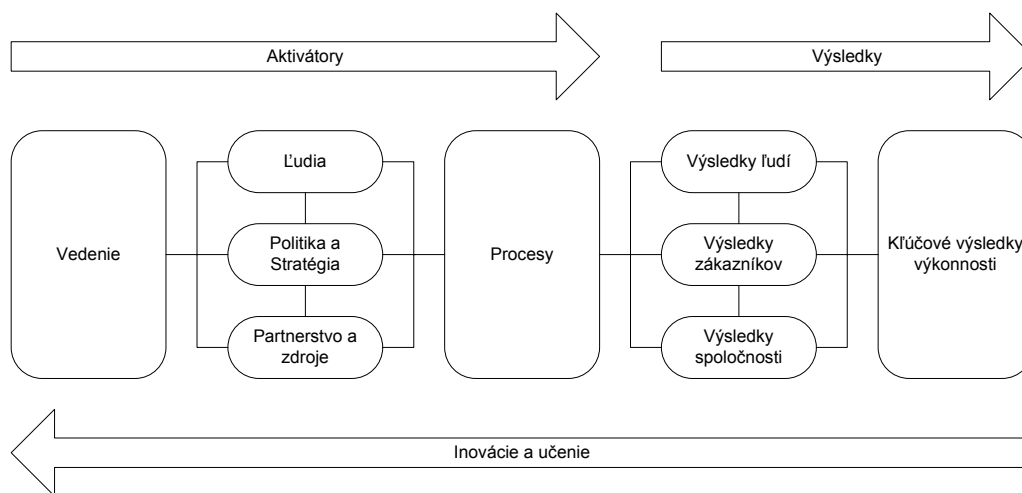
Niektoré modely definujú kvalitu procesov pre jednotlivcov, iné pre tímy a niektoré sú určené pre organizácie. Najznámejšie z týchto modelov sú:

- V Európe známy rámec pre organizačnú dokonalosť EFQM Model Dokonalosti, ktorý je podobný iným, napr. Malcolm Baldrige Model v USA;
- Dva štandardy pre procesy – ISO 9000:1994 a ISO 9000:2000;

- Skupina modelov IT zrelosti združená okolo CMM (Capability Maturity Model) a dva modely pre implementáciu CMM: TSP (Team Software Process) a PSP (Personal Software Process).

### EFQM Model Dokonalosti

EFQM (European Foundation for Quality Management) Model Dokonalosti [1] je organizačný model, ktorý používa nenormatívny rámec. Nie je to výslovne IT rámec. Môže sa použiť v organizáciách akejkolvek veľkosti a typu a je určený pre korporácie, spoločnosti alebo nezárobkové organizácie. EFQM poskytuje rámec pre dokonalosť v deviatich kritériách. Päť z nich sú tzv. “Aktivátory” pre dokonalosť a štyri sú metriky pre “Výsledky” (**Obr. 1**).



**Obr. 1.** EQFM Model Dokonalosti.

### ISO 9000:1994 and ISO 9000:2000

ISO 9000:1994 je štandard na zabezpečenie kvality v návrhových a výrobných procesoch. Tento štandard je prísny vo svojej definícii a interpretácii kvality. Definuje veľké množstvo procesov, ktoré treba dodržiavať s ohľadom na audity a evidenciu pričom nezáleží na tom, či tieto procesy poskytnú zákazníkovi alebo projektovému tímu ten najlepší výsledok. [2] V ISO 9000:2000 je menšie množstvo definovaných a dokumentovaných procesov a väčší dôraz na ľudí, aby porozumeli úlohám, ktoré musia vykonať a väčší dôraz na uspokojenie zákazníka. ISO 9000:2000 zahŕňa neustále zlepšovanie a približuje sa k EFQM. Hoci ISO 9000 štandardy spĺňajú niektoré Fundamentálne Konceptie Dokonalosti, sú v nich významné medzery, špeciálne v ISO 9000:1994.

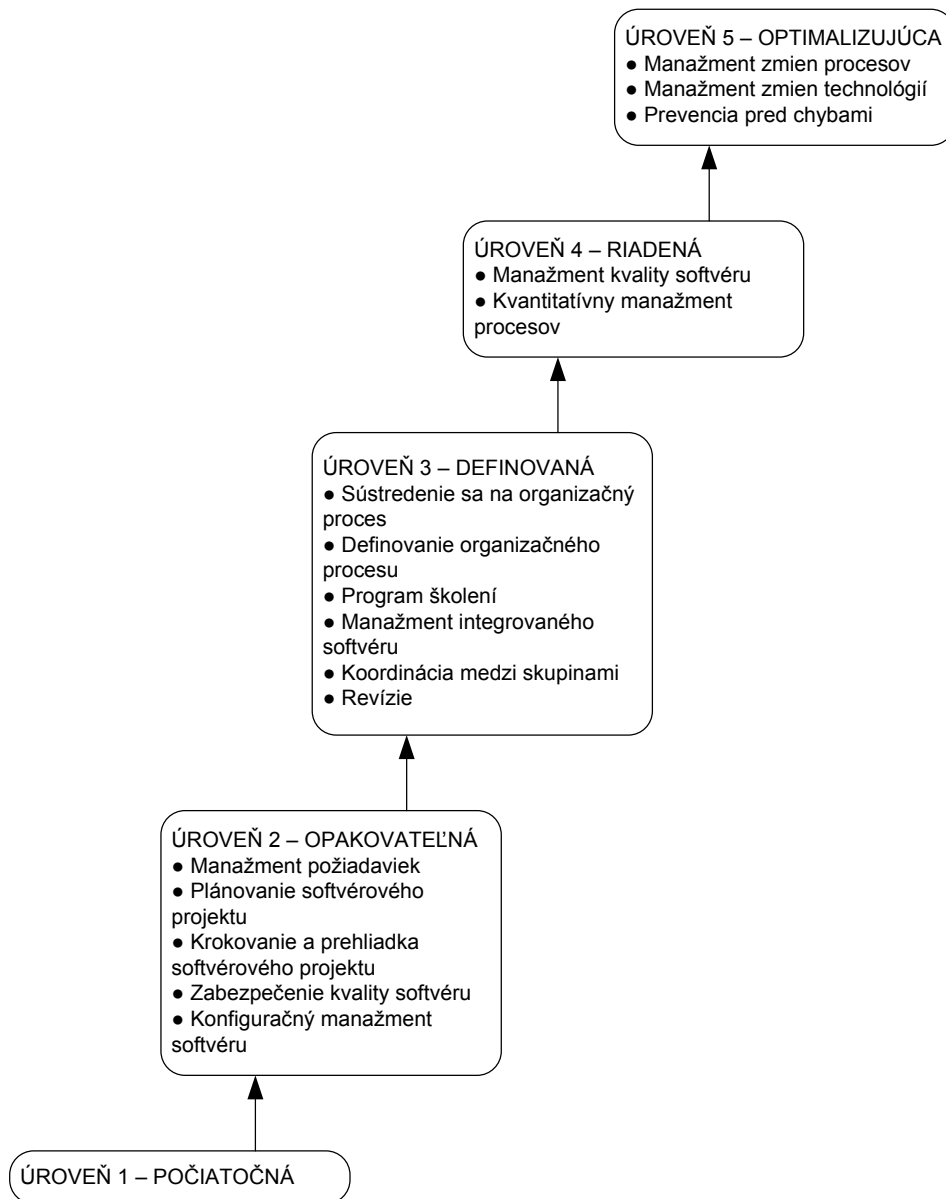
### Modely IT zrelosti – CMM a príbuzné modely

Koncept organizačnej zrelosti a spôsobilosti IT organizácií bol vyvinutý v SEI (Software Engineering Institute) a definuje lepší spôsob vývoja softvéru [1]. Softvér považuje za inžiniersku disciplínu a organizácie delí do piatich úrovní CMM (Capability Maturity Model) (**Obr. 2**) [3]:

- Úroveň 1 – Počiatočná: Projekty sú chaotické a ad hoc. “Každý má svoj vlastný proces.”
- Úroveň 2 – Opakovateľná: Požiadavky sú manažované a projekty sa realizujú podľa dokumentovaných plánov. “Každý tím má svoj vlastný proces; tímy môžu opakovať svoju prácu.”
- Úroveň 3 – Definovaná: Softvérové inžinierstvo a riadiace procesy sú stabilné a ostanú také aj pod stresom. “Každý tím v organizácii používa ten istý proces; môžeme sa začať zaoberať zmenami.”
- Úroveň 4 – Riadená: Organizácia riadi svoje procesy kvantitatívne a meria výkonnosť a kvalitu vo všetkých projektoch. “Celá organizácia sa meria, takže VIEME povedať ako sa nám darí.”
- Úroveň 5 – Optimalizujúca: Neustále zlepšovanie a aktívne riešenie nedostatkov. “Môžeme stavať na našom poznaní a ďalej sa zlepšovať.”

Pokrok cez tieto úrovne sa meria pomocou kľúčových indikátorov procesu. Všetky indikátory na jednej úrovni musia byť splnené predtým ako môžeme povedať, že organizácia je na tej úrovni. Úrovne a kľúčové indikátory sa sústreďujú na proces vývoja softvéru a meranie toho procesu.

CMM poskytuje odstupňovaný postup k zlepšeniu IT procesov. Základná premisa hovorí: IT organizácie potrebujú najprv kráčať predtým ako budú behať. Dômyselné inžinierske a meracie procesy nemôžu fungovať, pokiaľ nie sú postavené na systéme silných základných riadiacich praktík.



**Obr. 2.** Úrovne zrelosti v CMM.

CMM pokrýva vývoj softvéru a testovanie považuje za časť vývoja, pričom na testovanie kladie explicitné požiadavky na úrovni 3 a vyššie. Množstvo testerov v snahe zlepšiť túto oblasť začalo vyvíjať príbuzné modely špeciálne pre testovanie. Ťažkosti s implementáciou a používaním CMM [1] umožnili vzniknúť dvom ďalším procesným modelom: TSP (Team Software Process) a (Personal Software Process).

CMM a príbuzné modely sa stále vyvíjajú. SEI (Software Engineering Institute) nedávno uviedol CMMI (CMM Integration) a PCMM (People CMM) modely, ktoré sa pokúšajú rozšíriť aplikovateľnosť CMM koncepcií na všetky inžinierske disciplíny.

## Záver

V minulosti som pracoval v IT firme, ktorá nepoužívala žiadny štandard na zabezpečenie kvality softvéru. Podľa CMM teda dosahovala Úroveň 1. Procesy boli neprehľadné a neefektívne. To malo za následok zvýšené náklady na vývoj. V súčasnosti pracujem v IT firme, ktorá má už 4 roky akreditáciu na CMM Úroveň 2 a čoskoro by mala dosiahnuť Úroveň 3. Všetky procesy sú jasne definované a zdokumentované. Sice to vyžaduje od každého člena tímu zvýšene úsilie, ale to sa niekoľkonásobne vracia vo forme kvalitnejšieho softvéru a vyššej spokojnosti zákazníkov.

Odpoveď na otázku, či je manažment kvality dôležitý, znie jednoznačne áno a každá IT organizácia a firma, ktorá chce obstať na trhu v konkurenčnom boji, by mala implementovať niektorý zo štandardov na zabezpečenie kvality softvéru.

## Použitá literatúra

1. Evans, I.: *Achieving Software Quality through Teamwork*. Artech House, Norwood, 2004.
2. Gill, N.S.: Factor affecting effective software quality management revisited. In: *ACM SIGSOFT Software Engineering Notes*, Volume 30, Issue 2, ACM Press, New York (2005).
3. Jalote, P.: *Software Project Management in Practice*. Addison Wesley, Boston, 2002.

## Annotation

### *Quality management and the effect on the project result*

Developing a good software system is a very complex task. In order to produce a good software product, several measures for software quality attributes need to be taken into account. System complexity measurement plays a vital role in controlling and managing software quality because it generally affects the software quality attributes like software reliability, software testability and software maintainability. Thus, software quality assurance (SQA) needs to be addressed keeping in view the new strategies, tool, methodologies and techniques applicable to software development life cycle.

This essay is a survey study. Its main approach is to provide a compact view on the quality management in the software development process. The author answers the fundamental questions, what is quality management and why it is important. The author then compares existing models and standards for software quality assurance. In the conclusion, he describes the effect on the project result based on his practical experience.