

# KVALITA AKO ESENCIA MODERNÉHO PROJEKTU

*Kvalita nie je všetkým, ale bez kvality je všetko ničím.*  
Schoichiro Toyoda

Matej Podstrelenec

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
werther99[zavináč]gmail[.]com

**Abstrakt.** So slovom kvalita sa stretávame takmer každý deň. Počujeme o nej z prostredia obchodných reťazcov, výrobcov elektroniky, z automobilových závodov a čoraz častejšie sa s ňou stretávame i vo svete informačných technológií. Dnes tvorí kvalita dôležitý faktor v životnom cykle softvéru. Jej význam neustále rastie s počtom spoločností na trhu. V tejto eseji si pojem kvalita priblížime a stručne analyzujeme jej elementy definované metódou SQS z pohľadu vývoja softvérového projektu. Tieto elementy výraznou mierou zasahujú do životného cyklu každého systému a preto vyzdvihneme významné body každého z nich a pojednáme o príčinách ich zanedbania. V rámci eseje spomenieme i vybrané nástroje pre určovanie kvality, keďže vďaka nim dokážeme predísť nielen veľkému počtu chýb, ale vytvoriť i prostredie pre zabránenie ich replikácií. Na záver vyjadrím osobný postoj k elementom kvality metódy SQS a vyzdvihnem dôležitosť kvality ako esencie moderného softvérového projektu.

**Kľúčové slová:** kvalita, dôležitosť, SQS, manažér, element

## Úvod

Pri požiadavkách na softvérový projekt, popri aspektoch ako je čas alebo náklady, dozaista patrí i kvalita. Kvalitu môžeme definovať ako mieru, do ktorej množina základných charakteristík spĺňa požiadavky [3]. Táto skutočnosť pritom nezávisí od zložitosti projektu. V súčasnej dobe žiada koncový používateľ produkt, ktorého realizácia ho bude stáť čo najmenej, jeho spustenie bude možné uskutočniť ešte skôr, ako sa dosiahne dátum v zmluve a samozrejme bude poskytovať interakciu presne podľa predstáv. Dokonca ich ešte bude v mnohom prevyšovať. V reálnom svete však takéto príklady nevidíme často, pretože splniť spomenuté tri kritéria je veľmi náročné a žiada si tých najlepších ľudí vo všetkých odboroch. Preto sa zvyčajne naskytá otázka: „Čo urobiť, aby sa projekt stihol včas a neprekročil sa pritom rozpočtový limit?“. Kvalita pritom ostávala bokom. No práve v nasledujúcich odstavcoch si kvalitu predstavíme ako esenciu a teda základ moderného softvérového projektu.

Začať orientovať svoje zameranie na kvalitu môže byť spočiatku zložitý a časovo náročný krok, často odkázaný i na neúspech ešte predtým ako vôbec začneme. Podmieneny je nedostatočnou prípravou, nesprávnymi ľuďmi, nepresnými plánmi a mnoho ďalšími aspektmi. Je to ale kvalita, ktorej budovaním a obetovaním zdrojov dosiahneme náskok pred konkurenciou. Tak, ako to dokázalo japonské hospodárstvo po druhej svetovej vojne, keď americké spoločnosti nedokázali náporu kvality smerujúcej ponad Tichý oceán odolať. Aj keď produkovali spočiatku rýchlejšie a lacnejšie.

Paralelu môžeme nájsť i v softvérovom inžinierstve, kde dnes, po desiatkach rokoch úsilia sa stala kvalita popredným prvkom pri výbere, patriac medzi dôležité požiadavky projektu. Ak teda systém svojimi charakteristikami dokáže splniť tieto požiadavky, môžeme hovoriť o kvalitnej práci a o výhre kvality ako takej [1].

## Ako dosiahnuť kvalitu

Pre dosiahnutie kvality existuje niekoľko návodov a rokmi overených princípov. Podľa W. Horcha je dôležitá najmä metóda SQS (Software Quality System), ktorá je podmienená splnením nasledovných dvoch bodov.

Prvý bod sa dotýka budovania kvality od samého začiatku projektu. Treba teda zabezpečiť, aby navrhnuté riešenie mierilo na daný problém, rovnako ako správne porozumenie požiadavkám zadávateľa. Do doby, pokiaľ potreby a požiadavky nie sú podrobne známe a pochopené, je iba málo pravdepodobné, že dosiahneme výslednú spokojnosť založenú na kvalite.

Druhý bod zdôrazňuje nutnosť udržiavať kvalitu projektu počas celého životného cyklu. Toto znie ako neľahká úloha. S ohľadom na skutočnosť, že dodržiavanie zásad kvality so stúpajúcim časom má klesajúcu tendenciu, ostáva tento bod veľaokrát nespĺnený [2].

Pre splnenie spomínaných bodov a zaručenia kvality sa metóda opiera o nasledovných 10 kritérií:

1. Štandardy
2. Posudzovanie
3. Testovanie
4. Analýza zlyhaní
5. Manažment zmien
6. Bezpečnosť
7. Vzdelanosť
8. Správa dodávateľov
9. Ochrana
10. Riziko

Hore uvedené kritériá sú teda podľa SQS základom kvality projektu. O ich význame rovnako ako i dostatočnosti zabezpečenia kvality teraz stručne pojednáme. Zároveň sa tieto elementy kvality pokúsime doplniť i o vlastné skúsenosti získané prácou pre modernú softvérovú spoločnosť.

### **Štandardy**

Obdobie, keď bol softvér vyvíjaný bez toho, aby sa vývoj riadil normou alebo štandardom, je už pomaly isto za nami. Rôzne štandardy dnes zasahujú do každého úseku v životnom cykle vývoja projektu. Môžu opisovať spôsob zachytávania požiadaviek, použitia technológie a takisto aj formát samotnej dokumentácie. Šetria čas a zdroje spôsobom, že softvérový inžinier sa môže zaoberať viac technickými aspektmi projektu, ako zbytočne míňať energiu vymýšľaním už overených postupov na jeho tvorbu. Umožňujú sa teda viac zameriavať na samotný obsah, ako na detaily týkajúce sa formy. Každý manažér kvality by preto mal klásť dôraz i na prieskum využívania štandardov v tíme.

Z vlastnej skúsenosti viem ako prebieha tvorba systému, ktorá je riadená štandardmi. A ako prebieha tá, ktorá je ponechaná kreativite jednotlivých členov. Tá prvá má vysokú pravdepodobnosť dosiahnuť cieľ a splniť zadanie, pričom tá druhá zadanie často nespĺňa a navyše môže priniesť frustráciu do tímu pre neobjavenie tej správnej schémy vývoja.

Na druhej strane ale prílišné riadenie sa štandardmi môže paradoxne priniesť stratu rešpektu pred nimi a nesprávnemu použitiu štandardov. Preto by sa mala ponechať voľná cesta kreativite, kde špecifické postupy a metódy nie sú vždy absolútne potrebné.

### **Posudzovanie**

Posudzovanie napomáha k udržiavaniu viditeľnosti v systéme. Technický posudok je formálne, ale i neformálne preskúvanie častí projektu počas jeho vývoja v životnom cykle. Medzi neformálne posudky patrí i rekapitulácia, ktorej prejavom môže byť napríklad vysvetlenie funkcionality a zmysel komponentu ostatným zainteresovaným členom tímu ešte predtým, ako sa komponent stane základom pre ďalšie. Medzi formy posudkov patria i kontroly. Tie sú oproti rekapituláciám viac formálne.

Posudzovanie sa môže diať v ktoromkoľvek bode vývoja softvéru. Jeho hlavný zámer je preveriť splňanie cieľov projektu počas jeho vývoja. Kladie si za úlohu predísť nežiadúcim zlyháním a to prostredníctvom nástrojov kvality, ako napríklad Ishikavov diagram [4]. Diagram uvádza potenciálne hrozby a faktory, ktoré by sa pod budúci neúspech mohli podpísať. Preto pri vykonávaní posudku a pri narazení na nedostatky môže manažment rýchlo zasiahnuť a uviesť tak projekt na správnu cestu. Diagram je výnimočný svojim širokým využitím. Je ho možné aplikovať aj na náš vlastný život a predísť tak neskoršiemu neúspechu.

### Testovanie

Tento element patrí medzi základy kvality a jeho nedodržanie by mohlo mať za následok asi najväčšie škody. Úlohou testu je zvýšiť dôveryhodnosť systému a preukázať tak, že prvotne stanovené požiadavky na softvér sú napĺňané. Osobne považujem testovanie za tak dôležité, že ak opakovanie je matka múdrosti, tak test by mal byť rozhodne otcom budúcej kvality. Proces testovania začína už pri deklarácií požiadaviek až po dobu ich uskutočňovania. Každú požiadavku, ktorú nemožno otestovať, môžeme právom považovať za nesprávnu. Testovanie zohráva svoju najdôležitejšiu úlohu práve pri implementácií navrhnutého riešenia. Je priam esenciálne otestovať každú implementovanú časť skôr, ako sa začne implementovať nastávajúca. Nedodržanie testov sa v lepšom prípade odrazí neskôr pri implementácií nasledujúcej súčiastky. V tom neprijemnejšom prípade chybu objaví až samotný koncový používateľ, kedy náklady na jej odstránenie niekoľkonásobne prevyšujú náklady potrebné na jej skoršie otestovanie.

Pri práci na projekte som sa stretol so situáciou, keď jeho dokončenie bolo uponáhľané výročnou správou spoločnosti. Na projekte som pracoval spolu s ďalším kolegom a na úkor testovania sa nám podarilo prácu načas ukončiť, avšak za cenu zlyhania pri tretom použití nami vytvoreného systému. Výsledok bol pozastavenie obehu systému a pocit márnej snahy. Danú skutočnosť odvtedy beriem ako veľké ponaučenie o dôležitosti kvality, ako aj jej uprednostnenie pred časovým ohraničením.

### Analýza zlyhaní

Analýza zlyhaní predstavuje kombináciu zistenia zlyhania, korekcie a následnú analýzu vzniku samotného zlyhania. Ku zlyháním dochádza v každom softvérovom projekte. Ako som v predchádzajúcich častiach uviedol, ako jeden z nástrojov pre odhalenie slúži posudzovanie a kontrola. V praxi je postup nasledovný. Odhalí sa nedostatok, ktorý sa okamžite nahlási. Jeho neskoré nahlásenie, prípadne nenahlásenie, spôsobuje v neskoršom vývoji značné škody a obmedzuje tak výsledné dosiahnutie požadovanej kvality. Po nahlásení sa vytvorí vývojármí adekvátne riešenie, ktoré musí prejsť testami a byť odsúhlasené manažmentom zmien (Configuration Management). Úlohou manažéra kvality by teda mal byť aj dohľad, aby boli všetky zlyhania napravené a riadne zdokumentované. Treba vynaložiť úsilie pre ich budúce neopakovanie.

Splnenie tejto úlohy mu môže zaručiť i ďalší nástroj z prostredia kvality, diagram FMEA (Failure Modes and Effects Analysis). Diagram je zostavený ešte predtým, ako chyba, nedostatok vznikne. Má teda za úlohu predísť potenciálnym chybám a zabezpečiť

tak nerušený priebeh projektu počas vývojového cyklu. Použitie tohto prvku zlepšuje kvalitu produktu po všetkých stránkach [4].

Správny manažér kvality a nielen on, ale i celý vývojový tím, by mal klásť dôraz na neodkladanie riešenia vzniknutých problémov, nesnažiť sa opraviť nedostatky na úkor testovania a hlavne si zobrať ponaučenie a vzniknutej chybe sa v budúcnosti vyvarovať.

### **Manažment zmien**

Softvér, ktorý by neprešiel zmenami počas svojho vývoja, reálne neexistuje. Keď už teda príde čas zmeny, jej vykonanie je potrebné riadne označiť a zdokumentovať. Ak nie je známe, ktorá časť, komponent systému bola zmenená, môže nastať v tíme chaos, ktorý ovplyvní ďalší priebeh vývoja. Testy môžu byť aplikované na nesprávne verzie, vykonané zmeny môžu zase zasiahnuť už staršie a teda nepotrebné verzie systému. Preto systém bez kvalitného manažmentu zmien nedokáže uspokojiť zákazníka a vzniká hrozba, že bude pozostávať z nesprávnych komponentov.

Manažment zmien, ktorý okrem verziovania softvéru zabezpečuje i prevenciu neautorizovaných zmien v rámci produktu, má v začiatkoch životného cyklu primárny produkt dokumentáciu. Dôležitosť správy zmien rastie priamo úmerne s rozsahom, akým sa tieto návrhy stávajú skutočnosťou. Ak sa navrhne komponent s funkcionalitou čítania znaku, nebude sa bez príslušného odsúhlasenia a zdokumentovania vykonávať jeho zmena. Zmeny by mali byť špeciálne kontrolované v časti implementácie a testovania. Každá nová edícia dokumentu, súčiastky alebo komponentu dostane špecifický identifikátor a je zaznamenaná. Zo skúseností z tímového projektu, konkrétne tvorby tímovej stránky viem, že neverziovanie postupu pri jej tvorbe je pri viacčlennom tíme osudná chyba a okrem úpadku kvality sa plytvá i potrebným časom a zdrojmi.

### **Bezpečnosť**

Bezpečnosť sa týka nielen dát, ktoré je treba chrániť pred krádežou alebo modifikáciou, ale zaoberá sa aj ich fyzickou bezpečnosťou. Fyzickú bezpečnosť možno chápať ako ochranu pred vplyvmi prírody, ako sú záplavy, požiare a mnohé iné. Dáta je teda potrebné zálohovať na viacerých miestach súčasne. Pri zanedbaní povinností nepríde len ku nekvalite výsledného produktu, ale môže nastať situácia, že čas potrebný na dokončenie sa neúmerne navýši.

Kevin Mitnick uvádza, že i ten softvérovo najzabezpečenejší softvérový systém je možné prostredníctvom ľudského faktoru jednoducho penetrovať [5]. Preto by som dva atribúty bezpečnosti týkajúce sa dát samotných a ich fyzickej podoby doplnil i o bezpečnosť personálnych zdrojov. Sme to práve my ľudia, čo poskytujeme dáta tretím osobám, často bez overenia ich totožnosti. Pre podporu kvality by bolo preto vhodné zaradenie do procesu aktivít zamestnanca aj prípadné školenie o bezpečnosti a sociotechnike.

### **Vzdelanosť**

Vzdelanie zaisťuje, že členovia tímu vyvíjajúci softvér sú schopní vykonávať svoju činnosť korektné. Pre samotnú kvalitu je dôležité, aby vývojár ovládal viacero nástrojov podporujúcich vývoj. Programátor môže pociťovať ťažkosti, keď má za úlohu vytvoriť

projekt v C++, ak jazyk, ktorý pozná, je Visual Basic. Preto je potrebné, aby bol programátor s novým jazykom dostatočne oboznámený a neučil sa ho pri vývoji projektu. Kvalifikácia ovplyvňuje kvalitu nielen na úrovni vývoja, ale i samotnej prevádzky. Používateľ, ktorý ovládanie grafického editora pozná, dokáže vyprodukovať kvalitnejšie a predovšetkým rýchlejšie výsledky. A preto spojenie vzdelaných ľudí umiestnených na správnych miestach môže výrazne dopomôcť k úsiliu vyvíjanému pri tvorbe softvéru. Tento element, ako i mnohé predchádzajúce z metódy SQS, nepatrí primárne pod správu manažéra kvality. Keďže sa má však manažér zameriavať na všetky oblasti dotýkajúce sa kvality, rozhodne by i element vzdelanosť nemal obchádzať. S týmto elementom mám i osobnú skúsenosť a preto podporujem jeho dôležitosť. Pri práci na projekte, ktorý je úzko časovo ohraničený, je nesmierne náročné vyvinúť postačujúcu kvalitu a zároveň sa naučiť novú technológiu. Bez potrebných znalostí často dochádza k omylom a i k frustrácii vývojára. Preto by vzdelanosť mala tvoriť jeden z oporných pilierov kvalitného projektu.

### **Správa dodávateľov**

Ak pri tvorbe softvérového projektu vyberieme možnosť získania niektorých jeho častí od externých dodávateľov, stretávame sa s potenciálnym rizikom ohrozujúcim budúcu kvalitu. Takto získané komponenty možno rozdeliť na 3 časti. Ak zakúpime celý nový komponent, zaobstaráme komponent so základným rozhraním a o ďalšiu jeho funkcionálnosť sa postaráme sami, alebo zadáme požiadavky na vývoj externej spoločnosti. Prvá skupina je najkritickejšia, pretože kupujúci nemôže zaručiť, že vybraný produkt bude naozaj spĺňať potrebné meradlá kvality a robiť naozaj to, čo potrebuje. Druhá možnosť ponúka vyššie možnosti zabezpečenia kvality obstarávateľom. A to dôsledkom umožnenia aplikovania zásad kvality pre doplňujúcu funkcionálnosť vyvíjanú v domácom prostredí. Tretí spôsob zahŕňa vytvorenie celého systému treťou stranou, ktorej dodáme naše požiadavky. Výhoda tohto spôsobu je, že prístup ku kvalite zo strany dodávateľa môžeme uviesť priamo v zmluve. Je však potrebné, aby manažéri kvality na oboch stranách medzi sebou viedli úzku spoluprácu a vyhli sa tak neskorším nedorozumeniam, prípadným prekvapeniam. Špeciálny dôraz na externého dodávateľa by sa mal klásť aj keď pochádza z iného kontinentu, bez možnosti osobného kontaktu, ako napríklad stále rastúci obor informačných technológií India.

### **Ochrana**

Počítače zohrávajú čoraz dôležitejšiu úlohu v našich životoch. Dnes ich môžeme vidieť aj v medicíne, jadrových reaktoroch a vo vesmíre. Je preto nesmierne dôležité, aby sa počas životného cyklu softvéru kládol dôraz na ochranu budúcich používateľov. Tento element vystupuje do popredia najmä vo vyššie spomenutých oblastiach. Ak teda chceme produkovať kvalitu, nestačí sa len zamerať na splnenie požiadaviek. Je potrebné myslieť aj o jeho budúcom používaní. Prípadne zlyhanie, ktoré by sa podpísalo pod výbuch alebo iné životné ohrozenia, by nielen zničilo celé úsilie tímu, ale by rozložilo i celú spoločnosť. Preto každý softvérový projekt by mal zahŕňať dokument o otázkach ochrany a bezpečnosti.

## Riziko

Pod kvalitu každého softvéru sa bez pochyb podpisuje i manažment rizík. Jeho snahou je zabrániť vzniku možného rizika, ktoré by podmienilo nárast ceny, prípadne pozastavenie projektu. Je to predovšetkým zodpovednosť odboru rizík. Dobrý manažér kvality si dá pozor a pri udržiavaní potrebnej kvality sa bude snažiť dbať na zamedzenie vzniku akýchkoľvek rizík.

## Nechýba tam niečo ?

Pomocou metódy SQS sme v predchádzajúcich odstavcoch stručne predstavili jej elementy potrebné pre dosiahnutie kvality. Metóda zahŕňa skutočne aspekty, ktorých splnením môžeme vytvoriť projekt na veľmi dobrej úrovni.

Z osobného pohľadu by som ju však ešte rád rozšíril o ďalší element. A síce človek. Pravda je taká, že výsledná kvalita závisí od každého jednotlivca podieľajúceho sa na tvorbe softvéru. Jeho osobným prístupom dokáže kvalitu umocniť, ale ju i položiť na chrbát. Špeciálne by som však chcel vyzdvihnúť jeho vnútornú stránku. Člen tímu môže byť vzdelaný a ovládať znalosti potrebné na obsluhu nástrojov veľmi dobre. Ale jeho prínos nespočíva len v tom, ako obohatí samotný projekt, ale i v tom, ako obohatí ostatných členov tímu. Ak svojim správaním neudržiava potrebnú pohodu v tíme, vyrába konflikty, jeho akcie majú za následok znechutenie ďalších členov tímu a tým aj potenciálnu hrozbu pre kvalitu. Pokladám teda za veľmi dôležité pre kvalitu, aby tím pracoval v súlade a udržiaval medzi sebou dobré vzťahy.

Moje osobné skúsenosti sú poznačené práve týmto elementom. Ako každý nováčik v pracovnom živote sa toho musím veľa učiť. Je pritom základ pýtať sa na neznáme veci, ak nie je možnosť sa k nim inak dopracovať. Pri niektorých členoch tímu som sa však stretol viac s podráždenosťou, ako s reálnou pomocou. Na druhej strane pomoc, ktorá je dodaná s úsmevom na tvári pohne problém dopredu a aj pozdvihne morálku a entuziazmus pre boj s ďalšími problémami. Preto element človek a jeho pôsobenie na tím pokladám za rovnako dôležitý element kvality ako tie, ktoré boli opísané metódou SQS.

Neľahkú úlohu v tíme zohráva i samotný manažér kvality. Jeho cieľom by malo byť sa sústrediť čo najviac na zachovávanie spomenutých elementov a ich dodatočné obohacovanie o ďalšie. Avšak miera akou manažér vníma okolie je subjektívna. Mohli by sme ju pokojne vyjadriť myšlienkou: „Boh vidí všetko, manažér kvality bohužiaľ len to, čo vidieť chce“. Preto schopnosť manažéra hľadiť na problémy ovplyvňujúce kvalitu s rozvážnosťou a nezanedbávať ich, je nemenej dôležitá.

## Zhrnutie

Význam kvality neustále narastá aj vďaka zvyšovaniu počtu účastníkov na softvérovom trhu. Jej napredovanie môžeme sledovať nielen v oblasti softvéru, ale v službách ako takých a najmä v post-komunistických krajinách, kde pojem kvalita sa ešte nestihol naplno udomáčniť. Preto, ak chceme uspieť, mali by sme chápať kvalitu ako základnú esenciu každého a nielen softvérového projektu.

Cieľom tejto eseje bolo prezentovať znalosti o dôležitosti kvality a predstaviť metódu SQS, ktorá zahŕňa najpodstatnejšie elementy kvality týkajúce sa azda každého softvérového projektu. Tak, ako začal pred viac ako 50-timi rokmi budovať kvalitu v Japonsku William Deming, môžeme i my, vďaka SQS budovať kvalitu v našich tímových projektoch, rovnako ako i v celom našom živote.

## Použitá literatúra

1. Bieliková, M.: *Softvérové inžinierstvo. Princípy a manažment*. 1. vyd. Bratislava: STU, 2000.
2. Horch, J.W.: *Practical guide to software quality management*, London: Artech House, 2003.
3. ISO-9000:2000, Standards for quality management systems.
4. Mateides, A. a kol.: *Manažérstvo kvality*, Bratislava: Epos, 2006.
5. Mitnick, K., Simon W.: *Umenie klamu*, Bratislava: Helion, 2003.

## Annotation

### *Quality as a essential part of modern project*

*The word quality influences our everyday life. We can hear of quality from the domain of shopping centers, electronics corporations, car factories and even more often from the world of information technologies. Today, quality represents important value in software life cycle. Its importance grows with the number of companies on the market. In this essay, we are going to describe the idea of quality and briefly analyze its elements defined by SQS method in view of software project development. Those elements strongly affects software life cycle and therefore we will highlight most important points of each and shall discuss the situation of their neglect. Next we shall also take a closer look on chosen quality tools because they proved themselves in preventing many mistakes. In the end of an essay we shall express our attitude to SQS method and point out the importance of quality as essence of modern software project.*