

SOFTVÉROVÁ KVALITA NAŠA KAŽDODENNÁ

*Pochopiť nuly a jednotky nie je ťažké, ale kto v nich
vidí umenie, ten sa stane Picassom dnešnej doby,
ktorá ho nazýva programátor.*

Michal Kyžňanský

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
64384[zavináč]is.stuba.sk

Abstrakt. *Kvalita a najmä tá softvérová predstavuje magický pojem, ktorý od vzniku slova softvér trpí svojim nenaplnením. Každý manažment je v prvom rade o ľuďoch, a preto sa ako manažér kvality zamýšľam nad prístupom k nim, ako im byť bližšie a ťahať celý vývojový tím nenásilne na novú úroveň v otázke dosahovania kvality. Manažér kvality bojuje s veternými mlynmi, pretože bezchybný softvér je utópia a zaviesť nové postupy zlepšujúce kvalitu nie je ľahká úloha. Ego programátorov a zabehnuté kolaje sú len jedným z mnohých orieškov, ktoré musí manažér kvality rozlúsknuť. Zameriavam sa na postupy a správanie, ktoré pri správnej integrácii môžu zmierniť krízu kvality. Dôležitými zbraňami manažéra kvality je citlivý prístup, získanie si dôvery na všetkých frontoch a určovanie cieľov, ktorých naplnenie bude merateľné a zvýši kvalitu produktov nielen v očiach zákazníkov, ale dá aj vývojárom pocit dobre vykonanej práce. Vlastníci aj manažéri musia jasne vidieť benefity v podobe šetrenia nákladov do budúcnosti, aby bola misia kvalitéra úspešná.*

Kľúčové slová: *kvalita, softvér, manažment kvality, psychologické aspekty manažmentu*

Úvod

Softvérový svet plný rozsiahlych ambiciózných projektov trpí zaujímavými problémami, ktoré sú často spoločne nazývané pojmom softvérová kríza. Každý menší alebo väčší

projekt sa snaží dostať do magickej rovnováhy ceny a popritom vykorčuľovať z časovej tiesne, ktorá nevyhnutne doľahne takmer na každý vývojový tím. Je tu však ešte tretí aspekt, ktorý prísny rozpočet projektu spolu s ubúdajúcim časom okreše iba na splnenie základnej funkčnosti a odsunie kompletne na druhú koľaj. Áno, reč je o kvalite. Tej kvalite, ktorá by mala byť na prvom mieste, tej mýtologickej veličine, ktorá zahŕňa množstvo detailov. Kvalita je široký pojem a splnenie počiatočných požiadaviek predstavuje iba špičku ľadovca z vecí, ktoré by sme mali mať na pamäti pri tvorbe skutočne dobrého a kvalitného softvéru.

Vývojový tím obsahuje okrem najdôležitejších pozícií zvyčajne aj osobu, ktorá veľmi znepríjemňuje ostatným život a požaduje od nich "zbytočné" úsilie navyše. Je ale naozaj manažér kvality nepriateľ hrdých vývojárov a programátorov, ktorý si chce na svoj každodenný chlebič zarobiť znevažovaním práce tak cenných článkov vývojových tímov?

Na čo je dobrá kvalita?

Prečo sa vedú toľké diskusie o kvalite softvéru, možnostiach zlepšovania a vyvíjajú sa metodiky merania spoľahlivosti? Odpoveď je triviálna a ekonomická. Ide o čas a čas sú peniaze. Les Hatton sformuloval veľmi výpovedný zákon, ktorého znenie je skutočne alarmujúce. "*Ludia v technologickej spoločnosti vyhodia do vzduchu 250 dolárov na osobu ročne za softvér, ktorý nikdy neužije svetlo sveta, a ak náhodou áno, tak sa priblížil požiadavkám zákazníka iba málo*" (Hatton, 2007, s. 104).

Po zamyslení sa nad týmto tvrdením možno smelo konštatovať, že odvetvie tvorby softvérových systémov mrhá peniazmi mnohonásobne viac ako iné odvetvia, kde je efektívnosť práce verejne pranierovaná kvôli premrhaným nákladom.

Viacere príklady z vlastnej praxe mi len potvrdzujú slová, ktoré sformuloval Les Hatton. Pri menších projektoch, na ktorých som pracoval, sú problémy kvality takmer zanedbateľné, avšak nabaľovanie funkcionality a komplexnosť projektu jasne odhalia prvotné zanedbania a pochybenia v otázkach kvality.

Aj naoko dobre vyvíjajúci sa projekt postupným zavádzaním väčších či menších blokov funkcionality rýchlo degraduje v oblasti štruktúry a správy pokiaľ kvalita kódu, návrhu a spracovania nebola zahrnutá v počiatočných požiadavkách. Zvyčajne tento problém nastal, ak všetko úsilie bolo venované splneniu požiadaviek, ktoré kládli konkrétne prípady použitia. Ja ako vývojár som následne zažíval pocit odporu k vlastnej práci, pretože každý zásah do projektu, ktorý by ho mal posunúť z hľadiska klienta na novú úroveň funkcionality iba zvyšoval celkovú nestabilitu systému a robil jeho údržbu prakticky nezvládnuteľnou. Je teda ľahké si predstaviť, že projekt, do ktorého boli vložené veľké očakávania a najmä peňažné zdroje, skončí katastrofou, kedy už ani požiadavky klienta nie je možné zapracovať a projekt tím skončí v slepej uličke.

Vinníci sú všade naokolo

Každý programátor / vývojár je hrdý na svoj kód, svoju prácu, a preto pod farchou svojho ega často odmieta akékoľvek kontrolovanie svojej práce na najnižšej úrovni. Softvérová kvalita je z hry často vyšachovaná aj prístupom zvaným "nechaj to teraz tak" (Let-go) [2] a chyby sa budú opravovať neskôr. Firmy sa boja investovať do zvýšenej kvality softvéru,

pretože návratnosť tejto investície v žiadnom prípade nie je okamžitá. Ďalšie zlyhanie môže predstavovať aj nedostatok orientácie na delegovanie a dotiahnutie zmien spojených so zvyšovaním kvality do úspešného konca na všetkých úrovniach organizácie. Mýtom v tomto kontexte je aj idea, že ak sa na projekte pracuje dostatočne dlho, kvalita do neho akosi sama vsiakne [4].

Z vlastnej skúsenosti viem, že pri tvorbe rozsiahlejších webových systémov, ak už je produkt funkčný a nie je mu venovaná dostatočná pozornosť, sa v konečnom dôsledku po určitom čase používania skončí veľkými problémami. Typickým príkladom je požiadavka klienta novej funkcionality, ktorá skončí masívnym refactoringom a ten skonzuje mimo peňazí najmä predrahý čas programátora. V tej chvíli je vždy jasné, kde sa stala chyba, a že sa jej dalo predísť jednoduchým systematickým prístupom počas jednotlivých fáz vývoja. Kvalita, ako som už raz spomínal a budem na to apelovať, nepredstavuje úzku skupinu kvantifikovateľných atribútov, ale veľké množstvo drobných zmien či už priamo v práci alebo prístupe k problémom a ich riešeniu v rámci softvérového projektu.

Ak je potrebné ukázať prstom na toho kto je vinný, keď sa kvalita znižuje, tak podľa môjho názoru je v tom jednoznačne namočený celý vývojový tím i s manažmentom na najvyšších poschodiach.

V neposlednom rade ak hovoríme o kvalite softvéru, tak pri detailnejšom pohľade ide aj o kvalitu samotného návrhu v dlhodobom meradle. Hardvérové komponenty vykazujú zlyhanie vplyvom externých faktorov ako zmeny teploty a podobne, takže tieto zlyhania nesúvisia zvyčajne s návrhom produktu. V prípade softvérových riešení je možné veľké množstvo zlyhaní vystopovať až do návrhu jadra systému [1], čo v preklade znamená kódu samotného.

Manažér kvality alebo boj s veternými mlynmi

Manažér kvality má svoje miesto vo vývojových tímoch a je veľmi zaujímavé pozrieť sa na štatistiky, ktoré prináša hĺbková analýza na vzorke 50 renomovaných firiem ohľadom jeho pozície a najmä nazerania ostatných členov tímu na jeho úlohy [3].

Spriaznenou dušou v tíme je pre manažéra kvality od počiatku tester, ktorý si uvedomuje svoju rolu a vie, že zabezpečuje časť nutnej kvality, bez ktorej by produkt nemohol uzrieť svetlo sveta.

Na druhej strane, ak manažér kvality pochádza z prostredia manažérov, istotne v interpersonálnej komunikácii bude mať navrch, ale v konkrétnych aspektoch explicitnej kvality nebude zdatným oponentom vývojárov, ktorí ho tak môžu ľahšie presvedčiť, aby upustil od svojich požiadaviek.

Vedúci tímu sa zvyčajne sústreďí na úspech projektu ako celku. Často verí, že zvládnutím časového harmonogramu a udržaním projektu v rozmedziach rozpočtu sa kvalita dostaví viac menej sama, čo v konečnom dôsledku je mnohokrát opakom a nedostatky na poli kvality spôsobia posunutie termínov odovzdania [3].

Vývojári považujú každú úlohu navyše, ktorá nepredstavuje priamo proces dosiahnutia funkcionality, za bremeno. Pozná to iste každý, kto programoval nejaký systém. A taktiež sme všetci na svoj kód patrične hrdí. Tvorba dokumentácie a explicitné zvyšovanie kvality je považované za zbytočnú záťaž. Ak sa nájde aj uvedomelý vývojár, ktorý vidí prínos určitého refactoringu, často stojí pred schválením zo strany manažmentu,

ktorý vidí takéto úsilie ako bezpredmetné ak je projekt vo finále alebo už dokonca odovzdaný. [3]

Samotný manažér kvality má často iba poradnú úlohu a ak tomu tak nie je, nemá priamo vplyv na rozpočet projektu, takže sa musí uspokojiť s formálnym preukázaním kvality, čo podľa môjho názoru je zavádzajúci ukazovateľ [3].

Záver štúdie vypovedá, že uznávaný názor, kde manažment kvality je úplne odsúvaný na druhú koľaj nie je úplne pravdivý. Spoločnosti si dôležitosť a rolu tejto pozície vážia stále viac čomu možno v nemalej miere prispel aj fakt prechodu na agilné vývojové metódy. Avšak pohľady jednotlivých hráčov vo vývojovom tíme na explicitné zvyšovanie kvality jednoznačne hovoria, že manažér kvality má pred sebou ešte dlhú cestu a zatiaľ čaká na svoj rozkvet v oblasti softvérových projektov.

Zaujímavým meradlom úspešnosti manažéra kvality predstavuje pohľad na to z akej pozície sa prepracoval do pozície manažéra kvality. Podľa môjho názoru, ak pochádza z prostredia programátorov a vývojárov pracujúcich primárne so zdrojovým kódom, potom rozumie kvalite kódu, štábnej kultúre a dokáže explicitnú kvalitu v otázke udržateľnosti vývoja adresovať výborne. Taktiež jeho predchádzajúce pôsobisko môže mať vplyv na kolegov programátorov, ktorí ho budú brať ako jedného z nich a nie ako človeka, ktorý sa v problematike nevyzná a zbytočne ich poučuje. Avšak jeho slabou stránkou pravdepodobne bude komunikácia s inými pozíciami a osvetlenie prínosov práce na zvyšovaní kvality ako takej samotným stakeholderom.

Ako postupovať?

Organizácia a vývojový tím je predovšetkým zoskupenie ľudí, s ktorými treba pracovať citlivo, pristupovať k nim psychologicky a netváriť sa v prípade manažéra kvality ako absolútny odborník, ktorý despoticky nedáva priestor na diskusiu. Je nutné zapojiť priamo najnižšiu úroveň, programátorov a vývojárov samotných, aby postupne prichádzali aj s drobnými nápadmi alebo vylepšeniami, ktoré im v konečnom dôsledku spôsobia radosť a posunú celý projekt, aj keď po malých krôčikoch, dopredu v otázke kvality nenásilným spôsobom [2].

Myslím, si, že dobrý manažér kvality si musí získať rešpekt a dostať sa pod kožu celému vývojovému tímu. Ďalej si musí každý manažér kvality uvedomiť fakt, že bojuje s veternými mlynmi, t.j. softvér chyby jednoducho obsahuje. Je to objektívny fakt a na dosiahnutie absolútne bezchybného rozsiahleho softvéru by bol potrebný enormný čas, ktorý by hravo zhltoľ aj rozpočet niektorých menších štátov na svete.

Ďalší posun v očiach vývojového tímu je presvedčiť ho, že ste neprišli viniť ľudí za prehrešky v minulých projektoch, ale že váš pohľad je uprený do budúcnosti a chcete pracovať na kontinuálnom zlepšovaní v každom smere.

Existuje príslovie, že optimalizácia od počiatku je zlo (premature optimization is evil), avšak v prípade kvality je vždy dobré vedieť od začiatku ideálne ihneď po analýze, ktoré časti systému budú obsahovať doplnkové mechanizmy na zvýšenie kvality, aby sa s tým počítalo a vývojový tím vo fáze ukončovania projektu nemal pocit, že vyťahujete z rukáva požiadavky, o ktorých nikdy predtým nepočul.

Osobne si myslím, že je nutné dodávať tímu programátorov hmatateľné výstupy a výsledky, ktoré budú vypovedať o tom, ktoré konkrétne aktivity takpovediac navyše sú

zodpovedné za merateľné zvýšenie kvality celého produktu. Tieto informácie je rovnako nutné komunikovať aj smerom k vlastníkom (stakeholders), pre ktorých treba zvoliť zrozumiteľnú rétoriku s názornými príkladmi a najmä ukázať aspoň odhadovanú hodnotu úspor, ktorú dané akcie priniesli v dlhodobom meradle.

Ako ďalej štúdie ukazujú, pre naklonenie si manažérov a programátorov je veľmi vhodné používať ako presvedčovací nástroj konkrétne príklady zo zdrojového kódu, čo ukáže nielen kvalifikovanosť manažéra kvality v danom jazyku, ale aj ľahšie umožní získať pre tím viac prostriedkov na podobné procesy, ktorých cieľom je dosiahnuť vyššiu externú kvalitu.

Záver

Kvalita je veličina, ktorej je a bude venovaná stále väčšia pozornosť, pretože dôsledky jej zanedbania sú čoraz jasnejšie ako komplexnosť projektov všeobecne rastie. Úloha manažéra kvality sa stane jednou z najdôležitejších a posunie vývoj softvéru do nového svetla. Slovo kvalita v softvérovom svete je používané veľmi voľne, ale ide nám v skutočnosti o jednu jedinú vec. Aby sa úsmev zákazníka po prijatí a používaní softvéru vytrácal stále pomalšie, a aby sme my vývojári i manažéri kvality mohli spokojnejšie spávať s vedomím, že naše pracoviská opustil produkt, ktorý našim zákazníkom uľahčí život.

Použitá literatúra

1. Hatton, L.: The Chimera of Software Quality. In: *Computer*, vol. 40, 2007, no.8, pp. 104-103.
2. Nasib S. Gill: Factors affecting effective software quality management revisited. In: *ACM SIGSOFT Software Engineering Notes*, vol. 30, 2005, no.2, pp. 1-4.
3. Streit, J., Pizka, M.: Why software quality improvement fails (and how to succeed nevertheless). In: *Proceedings of the 33rd International Conference on Software Engineering*, Honolulu. 2011, ACM New York. ISBN 978-1-4503-0445-0, pp. 726-735.
4. Voas, J.: Software quality's eight greatest myths. In: *Software, IEEE*, vol. 16, 1999, no.5, pp. 118-120.

Annotation

Software quality in everyday life

Word quality and mainly the one referring to software represents a magical concept which unfortunately suffers from being unfulfilled from the very beginning of usage of word software. Every management mainly deals with the people and that is the reason why I as a quality manager focus on approach to the people, how to be closer to them and pull the whole development team in a nonviolent way to achieve new levels of software quality. Quality manager often experiences sciamachy because on the one hand flawless software is utopia and on the other introducing new practices that will raise the quality level is not an easy task. Programmers ego and old established

6 Michal Kyžňanský

practices are just a few of the many problems that are waiting to be solved by quality manager. I am focusing mainly on the procedures and behaviour which can result in reduction of problems connected to quality crisis if properly established and integrated. Important weapons that quality manager can use is sensible approach, gaining trust on every level of organisation and development team and setting objectives. Fulfilling of these objectives must be measurable and in the end will increase quality not only in front of customers but also give programmers sense of good work. To be successful as a quality manager owners and managers must clearly see the benefits brought by activities connected to quality management by means of reducing costs for future development.