

ROBME VECI (NE)KVALITNE

*Samotný manažment kvality nestačí na to, aby
vyvíjaný produkt bol kvalitný.*

Michal Sokolský

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

Abstrakt. *V súčasnej dobe existuje nepreberné množstvo programov a softvérových produktov, no mnohé z nich nedosahujú kvalitu, ktorú by si zaslúžili, resp. ktorá by bola žiadaná. Často dochádza k znižovaniu kvality už pri samotnom vývoji produktu. Vinu na tom majú viaceré faktory, s ktorými prichádza vývojový tím do styku. Manažment kvality poskytuje prostriedky na to, aby sa negatívny vplyv týchto faktorov minimalizoval. Pomáha využívať pri vývoji osvedčené metodiky a nástroje, ktoré do určitej miery zabezpečujú udržanie kvality na prijateľnej úrovni. Esej sa venuje niektorým závažným nedostatkom produktov z hľadiska kvality, ktoré sa veľakrát objavujú až po uvoľnení produktu do obehu. Hľadá príčiny vzniku týchto problémov a zamýšľa sa nad možnými riešeniami a nad tým, ako im je možné predísť.*

Kľúčové slová: *softvér, kvalita, faktor, produkt, manažment*

Úvod

Azda každý aspoň minimálne vzdelaný človek už vo svojom živote pracoval s počítačom. Či v práci, v škole, alebo doma vo voľnom čase. Vždy pracoval v nejakom programe, ktorý mu umožňoval vykonávať práve tú činnosť, ktorú potreboval. Umožňoval mu to viac či menej uspokojivo. Mnohokrát sa nám totiž stalo, že sme neboli schopní v programe vykonať istú operáciu, aj keď sme na tisíc percent vedeli, že program nám to umožňuje. Trošku sme si pospomínali na autorov aplikácie (veľakrát v zlom) a buď sa nám po vynaložení veľkej námahy podarilo vypátrať skryté tlačidlo, ktoré vykonávalo práve tú našu hľadanú činnosť, alebo sme si museli pomôcť radami iných ľudí. V najhoršom

prípade sme sa k riešeniu nedostali a pokračovali sklamaní ďalej, pokiaľ nás to natoľko neodradilo, že sme s daným programom nadobro skončili a našli za neho náhradu. Spôsob, akým sa s problémami vyrovnávame, závisí od kvality programu. Čím kvalitnejší produkt, tým menej problémov s ním máme. Počas vývoja produktov prechádza vývoj viacerými fázami. V pozadí vývoja sú aplikované mnohé podporné prostriedky, ktoré mu majú napomáhať a uľahčovať ho. Práve kvalite sa venuje manažment kvality.

Čo to je a na čo nám je manažment kvality?

Manažment kvality je spôsob zabezpečenia kvality produktu pomocou troch hlavných častí – kontroly kvality, zabezpečenia kvality a zlepšenia kvality. Na prvý pohľad nič zložité, ba príde nám to až samozrejmé. Manažment kvality možno aplikovať na rôznorodé činnosti, nie len na vývoj softvéru. Preto nám to je dôverne známe. Keď tvoríme nejaký produkt, tak chceme, aby bol čo najlepší, aby spĺňal všetky požiadavky, ktoré na neho budú kladené. Počas vývoja ho postupne skúšame, testujeme, či funguje správne (kontrolujeme jeho kvalitu), a ak zistíme nejaký nedostatok, snažíme sa o rýchlu nápravu (opravíme problém, čím zlepšíme jeho kvalitu). Niekedy nám to nestačí a sme príliš paranoidní. Vtedy sa dostávame do tretej časti manažmentu, kedy do nášho produktu zapracujeme opatrenia, ktoré majú zabrániť chybnéj funkcionalite nášho produktu (zabezpečujeme kvalitu). Tieto tri časti manažmentu kvality majú samozrejme svoju vlastnú vnútornú štruktúru a poskytujú rôzne prostriedky, ktorými sa snažíme dosiahnuť stanovený cieľ.

Zatiaľ to je fajn, v čom je teda problém?

Tajným snom všetkých analytikov, návrhárov, dizajnérov, či programátorov je, aby ich produkty boli čo najkvalitnejšie. Najradšej by ich azda priviedli do dokonalosti, ale ako všetci dobre vieme, nič nie je dokonalé a ani sa to nedá. Nebudeme si stavať vzdušné zámky. Na strane používateľov prevláda podobný duch. Chceme programy, ktoré budú pre nás ľahko použiteľné, budeme sa v nich bez problémov vedieť orientovať, intuitívne s nimi pracovať. Jedným slovom, budeme ich mať radi. Realita je však mnohokrát iná. Mnoho programov nedosahuje uspokojivú mieru kvality. Skrývajú sa za tým rôzne dôvody.

Spoločnosť Compuware uskutočnila v júli 2006 prieskum [2] v európskych IT firmách s cieľom zistiť, aký kvalitný softvér produkujú. Obdržala odpoveď od 150 firiem a výsledok bol šokujúci. Len 14% respondentov uviedlo, že majú zavedenú metodológiu riadenia kvality, ďalších 30% má formálnu metodológiu, no používajú ju len v niektorých projektoch. Viac než polovica organizácií má neočakávané problémy s výkonnosťou vo viac než 20% nasadzovaní svojich produktov. Po vydaní, uvoľnení produktu, bola väčšina organizácií v pozícii, kedy prekročili pôvodný rozpočet na produkt, dodali ho po termíne, alebo produkt nespĺňal očakávania používateľov. Aby toho ešte nebolo dosť, takmer 40% manažérov uviedlo, že v deň uvoľnenia aplikácie očakávali 50% pravdepodobnosť výskytu výkonnostných problémov. Svojim produktom naplno verilo len 5% manažérov. Pri zlyhaní produktu 39% organizácií pripisovalo vinu tímu testovačov a riadiacemu tímu. Ako sa hovorí, človek sa učí na vlastných chybách. Tu to však až tak úplne neplatí.

Približne polovica firiem čaká do výskytu prvých problémov počas testovania a vývoja, kým začne zbierať informácie potrebné k zabráneniu opakovania problému. 8% firiem sa tomu nevenuje vôbec. Zozbieraným výsledkom sa podrobne venuje a zapracúva len 16% firiem, približne polovica len v prípade potreby. Zvyšná časť ich buď nezberá, alebo sa s výsledkami nezaobrá. Pri takomto stave firiem sa nemožno potom čudovať fakt, že až 71% všetkých chýb je nahlasovaných samotnými používateľmi na linke technickej podpory. Ostatných 29% je odchytených IT pracovníkmi pomocou monitorovacích nástrojov.

Existuje riešenie?

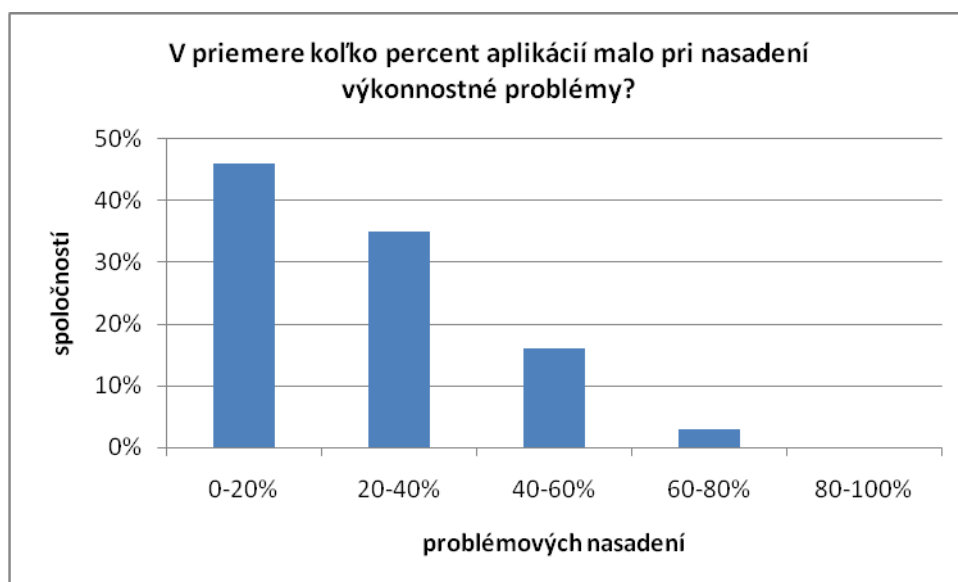
Samozrejme, že existuje. Na to manažment kvality predsa je, najmä časť venujúca sa zabezpečeniu kvality. Problém však nie je v manažmente ako takom, v metodológiách, postupoch a prostriedkoch manažmentu. Problém je v nás ľuďoch. Ľudský faktor je v mnohých oblastiach najkritickejší, ľudia nie sú stroje, nie sú neomylní. To je jeden z dôvodov, prečo sme nahrádzaní strojmi. Stroje sú schopné pracovať bez prestávky, stále s rovnakou presnosťou, nepotrebujú odpočívať. Keď sa nejaký pokazí, proste ho vymeníme za druhý. Netreba náročné školenia, či roky praxe, ktoré by človek musel mať, aby bol schopný nahradiť svojho predchodcu. V čom sa nám stroje zatiaľ nedokážu vyrovnávať, je myslenie. Robia len to, čo im prikážeme. Pri vývoji softvéru to však nestačí. Tam je potrebné logické myslenie, bez ktorého je nemožné urobiť použiteľný softvér. Už len pri špecifikácii problému, ktorý bude program riešiť, treba začať myslieť „ako by sme to asi tak mohli spraviť“. Logika človeka je skvelá vec, no často aj veľmi zradná. Môžeme si vymyslieť geniálne riešenie, no nedomyslíme veľakrát všetky jeho dôsledky, všetky prípady, ktoré môžu nastať. Keby sme boli stroj s logickým myslením, šlo by nám to oveľa jednoduchšie. Vtedy by sme vedeli ošetriť všetky situácie, do ktorých by sa bolo možné dostať. My ale ešte stále nie sme stroj, takže nastanú situácie, s ktorými sme jednoducho nepočítali a náš produkt na ne nevie správne reagovať a dochádza k chybe. Ale aby som nebol len negativistický, manažment kvality nám poskytuje prostriedky, pomocou ktorých je možné mnohým problémom zabrániť, vopred zabezpečiť správnosť a funkčnosť produktu, či počas vývoja nachádzať slabé stránky a chyby softvéru a následne ich odstrániť. Zabezpečenie kvality softvéru je založené na princípe prevencie. Naopak, testovanie (kontrola kvality) je orientované na detekciu problémov [3]. Zabezpečenie kvality sa opiera o praxou zavedené a odskúšané princípy, ako dosiahnuť čo najlepšiu kvalitu produktu. Existuje veľké množstvo štandardov, ktoré nás krok za krokom vedú celým vývojovým cyklom. Známy štandard ISO 9126 definuje 6 hlavných charakteristík kvality – funkčnosť, spoľahlivosť, efektívnosť, udržiavateľnosť, používateľnosť a prenositeľnosť [1]. Každá jedna časť je dôležitá, ak chceme mať kvalitný produkt.

Prečo k tým problémom vlastne dochádza?

Dovolím si tvrdiť, že vo väčšine prípadov sa za zlyhaniami a chybami v produktoch skrývajú tie isté príčiny. Ako už bolo spomenuté, ľudský faktor je veľmi častým pôvodcom chýb. Za mnohé chyby môže práve on. Od obvyčajnej únavy programátorov, cez nedostatočné znalosti v oblasti uplatnenia zamestnancov, nezáujem o prácu až

k nezmyselným termínom a z nich vyplývajúcemu časovému stresu pracovníkov, všetky tieto príklady a mnohé ďalšie sú dostatočne závažné na to, aby kvalita vyvíjaného produktu klesala závažným tempom. Aby som ale nehovoril len do vetra, poďme sa spolu pozrieť na niektoré problémy v spomínanom prieskume a pokúsime sa nájsť možné riešenie, ako sa dalo výskytu problému predísť.

Viac než polovica organizácií zaznamenáva problémy s výkonnosťou počas nasadzovania svojich aplikácií vo viac ako 20% (pozri Obr. 1). Problém je tu hneď zrejmý. Spoločnosti počas vývoja produktu priebežne netestujú svoj produkt z hľadiska výkonnosti, alebo sa tomu venujú len čiastočne. Každopádne to nie je v poriadku a mali by svoj postoj rýchlo zmeniť. Je potrebné zaviesť metodológiu, ktorá by definovala postupy kontroly produktu a zameriavala sa nielen na problémovú oblasť, čiže výkonnosť produktu, ale aj na ostatné oblasti, ktoré sú nemenej dôležité z hľadiska skvalitnenia výsledného produktu.

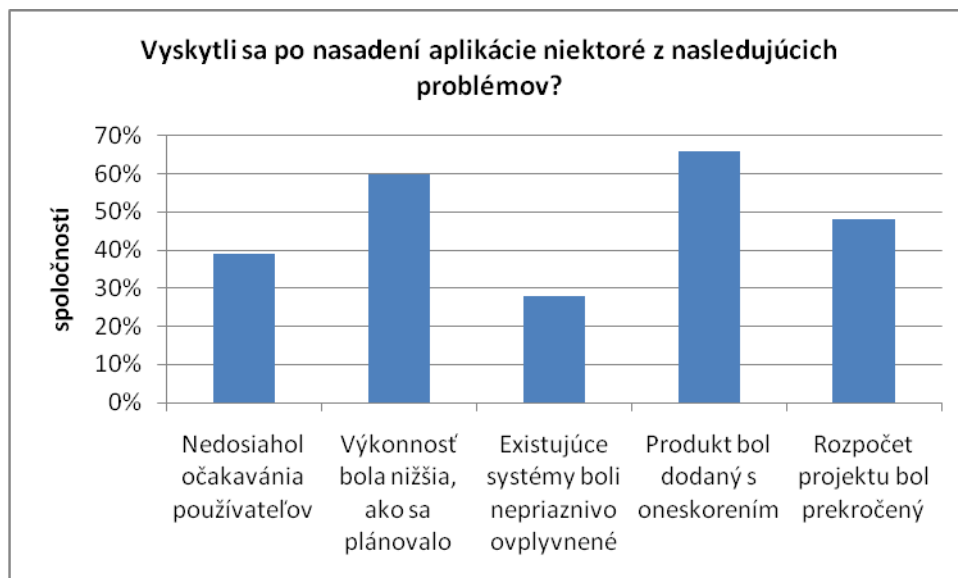


Obr. 1. Počet výkonnostných problémov pri nasadzovaní aplikácií.

Pre lenivejšie spoločnosti, prípadne spoločnosti, ktoré na tieto metódy nemajú čas (čo ich hneď stavia do zlého svetla, pretože nedbajú na kvalitu svojich výrobkov v dostatočnej miere), existuje ešte jedno alternatívne riešenie, ktoré ale v plnej miere nenahradí absenciu správnej metodológie. Riešenie je pomerne jednoduché a nevyžaduje si takmer žiadnu zmenu vo fungovaní spoločnosti. Dokonca sa dotýka len výkonných pracovníkov, ktorí pracujú na pozíciách priamo spätých s vývojom. Áno, reč je o programátoroch. Prečo práve táto skupinka? Odpoveď je veľmi jednoduchá. Tým druhým riešením je totiž zefektívniť zdrojový kód produktu natoľko, aby sa jeho náročnosť na prostriedky minimalizovala. Každému je jasné, že nie všetci programátori sú špičkoví experti, ktorí sú schopní písať len vysoko efektívny kód. Je to pekná myšlienka, ale v našom svete nereálna. Našťastie nepotrebujeme ani týchto expertov, ale vystačíme si s vlastnými zamestnancami. Keď ich firma zamestnáva, majú určité znalosti a schopnosti myslenia. K tomu sme

minimálne vedení v školách, aby sme sa naučili správne rozmyšľať nad problémami a vedeli ich vyriešiť tým najvhodnejším spôsobom. To nám stačí na to, aby sme programovali s rozvahou a vopred vedeli, čo je našim cieľom. Vopred mali premyslené, čo chceme. Potom sa dá rozmyšľať aj nad riešením, ktoré našu úlohu pojme čo najlepšie, najefektívnejšie. Najlepším riešením, ktoré môžu spoločnosti použiť, je zhostiť sa oboch spomínaných riešení. Zaviesť si formálnu metodológiu, ktorá zabezpečí kontrolu a priebežné sledovanie výkonnosti produktu a viesť svojich pracovníkov k tomu, aby produkovali čo najefektívnejší zdrojový kód. Je to malý krôčik k zlepšeniu spoločnosti, ale obrovský krok pre výsledné produkty.

Ďalším problémom spomenutým v prieskume boli konkrétne problémy, ktoré sa pri nasadení vyskytli. Väčšina spoločností uviedla viac než jeden problém, v priemere dva až tri zo spomenutých problémov na obrázku 2. Výsledky sú alarmujúce. Človek by ani neveril, že tieto problémy sú v úspešných IT spoločnostiach také časté. Poďme sa na niektoré pozrieť podrobnejšie. Výkonnostné problémy sme prebrali pred chvíľou, ostávajú nám na výber štyri.



Obr. 2. Vybrané problémy pri nasadení produktu.

Nedosiahnutá očakávaná kvalita produktu. Zlyhania možno hľadať v nedostatočnej špecifikácii produktu, chýbajúcej špecifikácii všetkých detailov. Často práve detaily najviac ovplyvnia koncového zákazníka, oni sú rozhodujúce, určujú, či sa produkt zapáči na prvý pohľad alebo nie. Nieкто by si povedal, že detaily nie sú dôležité. Práve, že sú. Detaily sa nemusia chápať len v rámci estetiky. S detailmi sa stretávame neustále, na detailoch stojí svet. Vráťme sa ale späť k problému špecifikácie. Je nevyhnutné, aby špecifikácia k projektu bola čo najpodrobnejšia. Inak nastane situácia, kedy sa pri vývoji narazí na problém, ako urobiť nejakú maličkú časť programu, ktorá nie je dobre opísaná. Vtedy programátor navrhne jedno riešenie, manažér druhé, ďalší kolega ešte jedno, no zákazník by to chcel práve tým spôsobom, ktorý oni všetci odmietli, lebo to nebolo

praktické alebo efektívne riešenie. No zákazník to tak chce. Ako sa hovorí, náš zákazník, náš pán. My mu jedine môžeme navrhnúť lepšie riešenie, ale ak sa on raz rozhodne pre to svoje, ťažko ho presvedčiť o opaku. Len si zahryzneme do jazyka a pokúsime sa zrealizovať jeho „skvelý nápad“. Druhou stránkou nedosiahnutia kvality je nechufť pracovníkov, ktorí na projekte pracujú. Robia len čo musia, ani čiarku navyše. Iné výsledky podáva človek, ktorého práca baví, ktorý jej rozumie, ako ten, čo pracuje viac-menej z donútenia. Takého človeka je potrebné správne motivovať a ak ani to nepomôže, nahradiť ho niekým iným a jeho zaradiť k inému projektu. V praxi nie je ojedinelé, že pracovníci (najmä programátori) majú averziu k istým veciam, napríklad programátor k niektorému programovaciemu jazyku, prostrediu, nástrojom a pracujú s nimi z donútenia a nechufou. Úlohou nadriadeného, resp. manažéra by malo byť, okrem iných povinností, mať istý prehľad o týchto slabých stránkach svojich podriadených a pri rozdeľovaní projektov brať ohľad aj na ne. Nepomôže tým len im, no dosiahne hlavne oveľa lepšie výsledky vo výslednom produkte.

Oneskorené dodanie produktu a prekročenie rozpočtu spolu veľakrát úzko súvisia. Počas implementácie sa vyskytne problém, ktorý treba následne riešiť. Lenže v časovom pláne sa s ním nepočítalo. Ak sa jedná o zložitejší problém, stratíme na ňom značné množstvo času, ktoré sa negatívne premietne do časového sklzu. Ten sa už veľmi ťažko doháňa. Navyše navýšenie času stráveného pri vývoji aplikácie si vyžaduje tento čas aj zaplatiť. Nie je možné si bez dôsledkov zmyslieť, že predĺžime vývoj o týždeň. Žiadny z našich pracovníkov nebude pracovať zadarmo. Pri plánovaní je potrebné plánovať s rezervou, dopredu počítať s komplikáciami. Mnohokrát sa nejedná len o technické komplikácie, pri rozsahovo menších projektoch (no nevylučujú sa ani veľké projekty) dochádza často k špecifikovaniu niektorých častí „za behu“, počas vývoja produktu. Veľkým problémom je aj zmena pôvodnej špecifikácie z iniciatívy klienta. Tu už nastávajú väčšie problémy, najmä ak je projekt v pokročilom štádiu vývoja. Nestáva sa to často, ale nemôžeme si zakrývať oči a tváriť sa, že sa to nikdy nestalo. Dopad na projekt závisí od miery, do akej zasiahne hotové a rozpracovaná časti produktu. Pri týchto častiach následne dochádza k duplicitným výdajom na ich vývoj. Raz sme ich už spracovali, ale z nejakého dôvodu ich musíme prerobiť. Často sa týmto zmenám nedá zabrániť, keď si raz klient niečo zaumieni, ťažko sa ho presvedča o opaku. Preto je lepšie radšej viac času stráviť na začiatku projektu s ním pri špecifikácii a dopodrobna sa všetkému venovať. Čím neskôr vstúpia do vývoja zmeny, tým bolestnejšie to pre projekt je.

Záver

Vývoj softvéru je sprevádzaný rozličnými vplyvmi, ktoré nepriaznivo vplyvajú na kvalitu výsledného produktu. Niektoré z nich je možné minimalizovať použitím správnych podporných prostriedkov, napríklad zavedením formálnej metodológie, dôsledným testovaním produktu počas vývoja a najmä pred jeho uvoľnením, či správnym plánovaním, ktoré počíta s komplikáciami počas vývoja. Samotný manažment kvality nestačí na to, aby vyvíjaný produkt bol kvalitný, mnohé ďalšie oblasti sa podieľajú vo väčšej či menšej miere na celkovej kvalite. Od výberu pracovníkov, cez dôkladnú špecifikáciu a naplánovanie vývoja, až po priebežné sledovanie a monitorovanie vývoja. Manažment kvality poskytuje prostriedky, ktoré nám umožňujú zastrešiť tieto oblasti a

napomáhajú k celkovému zlepšeniu produktu. Je len na každom z nás, či prijmem ponúkanú pomocnú ruku a dokážeme z neho vyťažiť maximum.

Použitá literatúra

1. Carvallo, Juan Pablo a Franch, Xavier. Extending the ISO/IEC 9126-1 quality model with non-technical factors for COTS components selection. *Proceedings of the 2006 international workshop on Software quality*. Shanghai : ACM, 2006, s. 9-14.
2. Compuware. Application Performance Management Survey by Compuware. [Online] Júl 2006. [Dátum: 12. 10. 2009.] Dostupné na internete: <http://www.docstoc.com/docs/425507/Application-Performance-Management-Survey-by-Compuware>.
3. Gill, Nasib S. Factors affecting effective software quality management revisited. *ACM SIGSOFT Software Engineering Notes*. 30, 2005, 2, s. 1-4.

Annotation

Let's make things (im)perfect

Nowadays there is amount of programs and software products, but many are not of quality they deserve, or quality required. Often it becomes to quality decreasing beside product developing. Several factors that the development team meets, cause such decreasing. Quality management is there to provide tools to minimize the negative impact of these factors. It helps to apply certified methods and tools used at developing, which guarantee the quality sustainment at the desired level. The essay devote to some significant defects of product in term of quality, which are detected after the product is released, mostly. It seeks for the causes of such problems and thinks of possible solutions as well as it tries to find how it is possible to beat them.