ZABEZPEČENIE KVALITY SOFTVÉRU V PROSTREDÍ MALÉHO TÍMU

Kvalita softvéru musí byť zabezpečená vždy a to bez ohľadu na veľkosť vývojového tímu.

Peter Študent

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

Abstrakt. *Vývoj softvéru je nesporne veľmi zložitý proces, ktorý je vďaka svojej komplexnosti nesmierne náročný na zaistenie kvality výsledného produktu. Kvalitu softvérového produktu môžeme vnímať z viacerých hľadísk, či už z hľadiska funkčnosti, použiteľnosti, výkonnosti alebo spoľahlivosti. Záujmom zákazníka, pre ktorého sa softvér vyvíja, je pritom dostať za svoje investované financie čo najkvalitnejší softvér po všetkých stránkach. Tento dôraz na kvalitu je zo strany zákazníka rovnaký bez ohľadu na to, či je obstarávaný softvér svojim rozsahom veľký a na jeho vytváraní sa podieľa rozsiahly tím, alebo či ide o menší projekt, vytváraný užším tímom. Veľké vývojárske tímy majú zväčša vyčlenený určitý počet svojich členov, ktorých úlohou je počas celého životného cyklu permanentne sledovať kvalitu vyvíjaného produktu a takto napĺňať očakávania zákazníka v oblasti kvality. Malý vývojový tím sa musí z tohto pohľadu uskromniť, čo však určite neznamená, že by mal menej dbať na kvalitatívne kritéria vyvíjaného produktu. Cieľom tejto eseje je oboznámiť čitateľa s problematikou testovania softvéru v prostredí malého vývojového tímu a načrtnúť riešenia, akým spôsobom sa dá aj v prostredí malého vývojového tímu vytvárať softvérový produkt s vysokými kvalitatívnymi parametrami.*

Kľúčové slová: *testovanie softvéru, kvalita softvéru, vývoj riadený testami*

Kvalita softvéru a jej dopad na spoločnosť

Ľudstvo dnes žije v storočí, ktoré sa dá nesporne charakterizovať ako storočie informačných technológií. Bez osobného počítača, prístupu k elektronickej pošte, či najnovším správam zo sveta si vie dnes veľká časť populácie predstaviť svoj život len ťažko. Počítačové systémy nás obklopujú na každom kroku a celé hospodárstva vyspelých štátov sveta sú bezpodmienečne závislé na informačných technológiách.

Vďaka informatizacii a automatizácii výrobcovia produkujú obrovské množstvá výrobkov v nebývalej kvalite, distribučné spoločnosti ich masívne presúvajú k spotrebiteľom a spotrebitelia si užívajú široký sortiment produktov s priaznivými cenami dosiahnutými hromadnou výrobou. Výrobcovia vďaka presným údajom o predajoch vedia upravovať skladbu svojho sortimentu a efektívne reagovať na požiadavky trhu a to je len zlomok toho, čo nám priniesla táto informačná epocha. Nie sú to pozitíva len v ekonomickej oblasti, ktoré nám priniesol rozvoj informačných technológií. Informačné systémy denne pomáhajú v oblasti zdravotníctva, bezpečnosti a v množstve iných, pre spoločnosť nesmierne dôležitých oblastiach. Celkovo možno konštatovať, že závislosť spoločnosti od informačných technológií je momentálne na takej vysokej úrovni, že zlyhania jednotlivých informačných systémov môžu mať ďalekosiahle dôsledky.

Žiaľ, ľudstvo už zažilo viacero podobných zlyhaní, ktoré buď zapríčinili veľké ekonomické škody, ba čo je horšie, v niektorých prípadoch aj straty na ľudských životoch. Takýmto zlyhaním bola napríklad softvérová chyba lekárskeho prístroja Therac-25, ktoré slúžilo na počítačovo riadenú rádio terapiu [4]. Chyba v softvéri tohto zariadenia zapríčinila to, že v rozmedzí rokov 1985 až 1987 pri najmenej šiestich prípadoch došlo k výraznému prekročeniu dávky žiarenia, čo priamo zapríčinilo smrť niekoľkých pacientov.

Podobné incidenty vyvolávajú medzi odbornou ale aj laickou verejnosťou otázniky nad kvalitou softvéru, na ktorom sme tak životne závislí. V tejto súvislosti by asi bolo vhodné položiť si otázku, čo máme rozumieť pod pojmom kvalita softvéru? Možno sa vám zdá táto otázka svojou podstatou primitívna, avšak zodpovedanie tejto otázky nie je vôbec také triviálne, ako sa na prvý pohľad zdá.

Na pojem kvalita v softvérovom systéme sa nedá pozerať len z jednej perspektívy. Každý zainteresovaný účastník na životnom cykle softvérového systému vníma kvalitu softvérového produktu odlišne. Pri vnímaní kvality totiž záleží na tom, akú rolu hrá daný účastník pri vývoji, resp. prevádzke softvérového systému. Používateľ sa asi na rozdiel od vývojára nebude pýtať "Je kód tohto systému dobre štruktúrovaný?", ale skôr "Pomáha mi tento softvér odviesť viac práce?", presne tak, ako to vo svojej úvahe uviedol Peter J. Denning [1].

Otázka teda znie, ktorý z týchto pohľadov na kvalitu je ten správny? Vychádzajúc z definície pojmu kvalita, uvedeného v norme ISO 9001-00, pod kvalitou rozumieme "mieru, ktorou základné charakteristiky spĺňajú definované požiadavky" [6]. Táto definícia je zvolená natoľko široko, že zahŕňa pohľad všetkých zainteresovaných strán. Ako vyplýva z definície, je nesmierne dôležité pre sledovanie kvality vopred jasne a presne zadefinovať požiadavky kladené na výsledný produkt a charakteristiky, pomocou ktorých budeme jeho kvalitu hodnotiť.

Bežné postupy na zabezpečenie kvality softvéru

Problematike kvality softvéru sa doteraz venovalo a naďalej venuje množstvo rôznych štúdií a výskumov. Postupne sa vyčlenila samostatná oblasť manažmentu kvality, ktorá nesie názov zabezpečenie kvality softvéru (z ang. software quality assurance). Vo veľkých firmách sa bežne stretávame s tým, že sa o zabezpečenie kvality vyvíjaného softvérového systému starajú špecializované tímy ľudí, ktoré nie sú zložené z programátorov priamo vytvárajúcich kód aplikácie, ale členovia týchto tímov majú na starosti len samotný proces zabezpečenia kvality. Na kvalitu testovania v takýchto samostatných tímoch pritom nevplýva ani tak množstvo zapojených ľudí do procesu testovania, ale skôr skutočnosť, či sú tieto tímy zložené zo skúsených odborníkov, ktorí veľmi dobre poznajú problémovú oblasť.

V prostredí malých vývojových tímov je však veľmi problematické vzhľadom na obmedzené finančné alebo ľudské zdroje zabezpečiť vytvorenie takých špecializovaných pracovných pozícií, ktoré by sa starali výhradne len o zabezpečenie kvality výsledného softvérového produktu. Nehovoriac už o možnostiach získať človeka s expertnými znalosťami z problémovej oblasti. Znamená to teda, že malé vývojové tímy produkujú menej kvalitný softvér ako veľké spoločnosti? Takéto zjednodušenie rozhodne nie je na mieste. Kvalita softvéru musí byť zabezpečená vždy a to bez ohľadu na veľkosť vývojového tímu.

Zabezpečenie kvality v malom vývojovom tíme

Aj napriek obmedzeným zdrojom v prostredí malého vývojového tímu existuje množstvo metód, ktoré napomáhajú zabezpečiť kvalitu softvérového produktu. V tejto súvislosti je potrebné sa zamyslieť nad vzťahom medzi internými a externými charakteristikami kvality softvéru, tak ako tieto pojmy zadefinoval Steve McConnell. Podľa jeho definície pod externými kvalitatívnymi charakteristikami rozumieme tie časti produktu s ktorými prichádza užívateľ priamo do styku a medzi interné sa zaraďujú tie, kde táto interakcia priamo nenastáva [5].

Je nespochybniteľné, že interné charakteristiky kvality softvéru majú dopad aj na externé charakteristiky kvality, pričom opačne táto implikácia platiť nemusí. Ako jedno z východísk v oblasti zabezpečenia kvality softvérových produktov v prostredí malých vývojových tímov vidím kladenie vyššieho dôrazu práve na tieto interné kvalitatívne charakteristiky. V súčasnosti existuje viacero prístupov, ktoré sa zameriavajú práve na tieto interné kvalitatívne charakteristiky. Ich nasadenie si pritom nevyžaduje dramatický nárast finančných alebo personálnych zdrojov a práve preto ich nasadenie v malých vývojových tímoch býva relatívne bezproblémové.

Softvérové inšpekcie

Medzi prístupy , ktoré sa zameriavajú a interné kvalitatívne charakteristiky softvéru, sa dajú zaradiť aj softvérové inšpekcie, ktoré vychádzajú z princípu "viac očí viac vidí". Ich podstatou je vzájomná kontrola jednotlivých výstupov člena vývojového tímu iným členom tohto tímu. Softvérové inšpekcie by sa pritom nemali obmedziť len na samotný zdrojový kód, ako sa to často stáva. Oveľa lepší výsledný efekt na kvalitu softvéru budú

mať tieto inšpekcie vtedy, ak sa budú aplikovať aj na iné ľuďmi čitateľné výstupy z vývoja softvérového produktu. Vhodnými kandidátmi na zameranie sa softvérovej inšpekcie sú napríklad požiadavky, špecifikácie, modely, dokumentácie a iné dokumenty. Softvérové inšpekcie sú pritom jedny z mála metód, ktoré sa dajú aplikovať hneď od prvotných fáz vývoja softvérového produktu. Medzi odbornou verejnosťou je pritom všeobecne známy fakt, že čím neskôr sa chyba vo vyvíjanom softvéri objaví, tým je potrebné na jej odstránenie vydať väčšie úsilie. Inými slovami, čím neskôr sa zistí chyba v softvéri, tým stojí firmu viac vynaložených finančných prostriedkov.

Z vlastnej skúsenosti môžem povedať, že softvérové inšpekcie prinášajú do vývojového tímu aj ďalšie pozitívne efekty. Jedným z nich je aj to, že svojou podstatou napomáhajú prirodzene prenášať vedomosti a skúsenosti od skúsenejších členov tímu k tým menej skúseným a tým do určitej miery nahrádzajú rôzne školenia pre menej skúsených vývojárov. Ďalším nesporne pozitívnym efektom je aj pozitívny psychologický dopad na členov tímu, keďže softvérové inšpekcie napomáhajú komunikácií v rámci tímu a predchádzajú konfliktom.

Niektoré metodiky, obzvlášť tie agilné, majú softvérové inšpekcie priamo zakomponované vo svojich postupoch. Príkladom môže byť extrémne programovanie [3], kde sa tento postup praktikuje v takzvanom párovom programovaní. Nasadenie týchto postupov samozrejme stojí určité finančné prostriedky. Odhaduje sa, že na celkových nákladoch na vývoj sa môžu pohybovať v rozmedzí od 5 do 15 percent [7]. Aj keď sa to môže zdať veľa, tieto náklady sú zanedbateľné v porovnaní s ušetrenými nákladmi na následnú údržbu softvérového systému.

Vývoj riadený testami

Ďalším prístupom, ktorý podľa mňa môže výraznou mierou prispieť k zvýšeniu kvality výsledného softvérového produktu, je vývoj softvéru riadený testami. Táto technika vývoja softvéru pozostáva z krátkych vývojových iterácií, ktoré sa ďalej skladajú z dvoch základných krokov. V prvom kroku sa vytvárajú podľa špecifikácie testy, ktoré pokrývajú elementárnu funkčnosť. V ďalšom kroku sa potom pristupuje k samotnej implementácií danej funkčnosti, ktorej výsledkom by malo byť úspešné absolvovanie všetkých dovtedy napísaných testov. Následne sa ešte môže v prípade potreby pristúpiť k refaktorizácii zdrojového kódu.

V odbornej verejnosti veľa krát dochádza k diskusii, aký vplyv má zmena poradia týchto dvoch krokov na produktivitu a výslednú kvalitu softvérových produktov. Je vhodnejšie písať najprv test a až potom kód alebo opačne? Podľa štúdie [2] je produktívnejší práve prístup pri ktorom sa vytvárajú testy napred. Vyššiu produktivitu autori pripisovali najmä lepšiemu porozumeniu zadaniu, sústredeniu sa len na jednu úlohu a menšej námahe pri opravách chýb, ktoré tento prístup zo sebou prináša.

Testy môžeme podľa ich zamerania zaradiť do dvoch hlavných skupín. Jednotkové testy kontrolujú, či jednotlivé komponenty fungujú správne a teda či implementácia vyhovuje špecifikácii. Akceptačné testy sa v porovnaní s jednotkovými testami nezameriavajú na jednotlivé detaily systému, ale naopak skúmajú ho ako jeden celok. Na základe scenárov použitia, ktoré vychádzajú z požiadaviek zákazníka, pre ktorého sa

softvér vyvíja, sa vytvoria jednotlivé akceptačné testy. V prípade, že sa podarí úspešne prejsť cez všetky akceptačné testy, vývoj softvéru sa považuje za dokončený.

Pomerne častým javom pri zavádzaní techniky vývoja riadeného testami, s ktorým som sa aj ja stretol, je určitá nechuť programátorov písať testy. Zvyčajne argumentujú tým, že tvorbou testov by strávili toľko času, že by prestali stíhať plniť plán vývoja. Tento pohľad je ale krátkozraký a vedie do začarovaného kruhu. Čím totiž programátor píše menej testov, tým je aj menej produktívny, nakoľko jeho kód je menej stabilný a musí venovať viac času opravám svojho kódu. Čím programátor venuje viac času opravám, tým mu ostáva menej času na písanie testov. A sme znova na začiatku. Pri nasadzovaní techniky vývoja riadeného testami je potrebné byť trpezlivý a dôkladným vysvetľovaním je treba prekonať počiatočný odpor. Keď si programátori na techniku vývoja riadeného testami zvyknú, bude im pripadať ako samozrejmá a sami nebudú rozumieť tomu, prečo ju nezačali používať skôr.

Obr. 1. Postupnosť krokov pri vývoji riadenom testami.

Veľmi dôležitým prínosom vývoja riadeného testami je jeho pozitívny vplyv na udržovateľnosť zdrojového kódu. Veľa krát sa v praxi stretávame s tým, že vývojári softvéru neradi robia úpravy v starom kóde aplikácie, lebo sa boja, že do aplikácie zanesú nové chyby, resp. že pri pridaní nového modulu prestane správne fungovať niektorý z pôvodných modulov. Tento jav popisujú aj pokročilí užívatelia, keď položartom pri inštalácií aktualizácie softvéru skonštatujú, že sa im namiesto opráv chýb, inštalujú nové chyby. V prípade, že zdrojový kód je pokrytý testami sa možnosť vzniku nových chýb eliminuje. Vývojári softvéru sú si vedomí toho, že ak by aj úpravou kódu niečo znefunkčnili, zásluhou vývoja riadeného testami by vedeli, kde nastal problém a nemuseli by dlho pátrať po jeho príčinách.

Pomerne veľkým nedostatkom testovania programátormi, s ktorým som sa už viackrát v praxi stretol, je príliš jednostranný pohľad na vyvíjaný softvér. Nakoľko autormi testov sú samotní programátori, veľa krát im uniká pohľad bežného užívateľa, čo sa prejavuje najmä v používateľskej „prítulnosti“ aplikácie. Programátori, a teda tvorcovia

testov, sú pri vývoji aplikácie od jej počiatku, takže zvyčajne sa za ten čas s ňou zžijú natoľko, že postupy, ktoré by sa zdali bežnému používateľovi príliš zložité, oni berú ako samozrejmé. V konečnom dôsledku to má za následok to, že aj keď softvér prechádza cez všetky tieto automatické testy, môže byť pre koncového používateľa len ťažko použiteľný.

Kvalita softvéru na dosah ruky?

Je teda schopný aj malý tím dostatočne zabezpečiť výslednú kvalitu vyvíjaného softvéru? Stačia vyššie uvedené prostriedky na jej zabezpečenie? Nie je v konečnom dôsledku potrebný aj samostatný testovací tím? Podľa môjho názoru má aj malý tím k dispozícii dostatok nástrojov na zabezpečenie kvality vyvíjaného softvéru, z ktorých niektoré som spomenul v tejto eseji. Tieto prostriedky už v súčasnosti umožňujú malým tímom dosahovať dobré výsledky v oblasti kvality. Nevyžadujú si pritom výrazné navýšenie nákladov na vývoj softvéru, alebo potrebu vyčleňovať časť členov tímu len pre potreby testovania.

Samozrejme, prostriedky, ktoré som uviedol, sa zameriavajú na internú kvalitu softvérového produktu, pričom stále zostáva určitá oblasť, ktorej kvalita sa nedá zabezpečiť len prostredníctvom uvedených prostriedkov. Ide oblasť externých softvérových charakteristík kvality, aj keď ako som v úvode uviedol, vylepšenie interných softvérových charakteristík kvality má pozitívny vplyv aj na ne. Vyššie uvedené prostriedky na zabezpečenie kvality vyvíjaného softvéru ako je vývoj riadený testami alebo softvérové inšpekcie nie je možné preceňovať a je potrebné byť si vedomý ich limitácií, pretože netvoria stopercentný nástroj na zaručenie vysokej kvality softvérového produktu, ale určite ho k tomuto cieľu výrazne približujú.

Použitá literatúra

1. Denning, P.J.: *Editorial: What is software quality?* Commun. ACM 35, 1 (1992), 13-15.
2. Erdogmus H., Morisio M, Torchiano M.: On the effectiveness of the test-first approach to programming. *Proceedings of the IEEE Transactions on Software Engineering*, 31(1). (2005)
3. Extreme Programing: http://www.extremeprogramming.org, [cit: 2009-October]
4. Leveson, Nancy G., and Turner, Clark S. (July 1993). *An Investigation of the Therac-25 Accidents* Computer (IEEE), 18-41.
5. McConnell, Steve (1993), Code Complete (First ed.), Microsoft Press
6. TC 176/SC (2005). ISO 9000:2005, *Quality management systems – Fundamentals and vocabulary*. International Organization for Standardization.
7. Wiegers, K.E.: *Improving quality through software inspections*. Software Development, vol. 3, no. 4 (1995)

Annotation

Software quality assurance in a small development teams

Software development is undoubtedly a very complex process, which is due to its complexity extremely difficult to ensure the quality of the developed product. Software product quality can be seen in many ways, whether in terms of functionality, usability, performance and reliability. Customer interest is to get over their invested funds, high-quality software in all aspects. This emphasis on the quality of customer is the same regardless of whether their software is procured a large scale and its implementation involved a large development team, but whether it is a smaller project, created by the smaller team. Large teams are generally allocated a certain number of its members, whose role is during the lifetime of the permanent monitoring of the quality of developed product, and thus meet customer expectations for quality. A small team must be from this perspective draw in, but it certainly does not mean that it should be less sure of the quality criteria, quite the contrary. The aim of this essay is to inform the reader about software testing issues in a small team environment and outline the solution, how can even a small team environment to create software products with high quality parameters.