

KAŽDÝ ROBÍ CHYBY...PRETO MÁME TESTEROV

Priatelia sa neboja poukázať na svoje chyby.

Adam Pomothy

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

Abstrakt. *Tester sa často považujú za jednorazových a dočasných zamestnancov, ktorým sa dajú k dispozícii len testovacie scenáre a pravidlá, ako hlásiť nájdené chyby. V skutočnosti ale tvoria rovnocenných partnerov vývojárskeho tímu so spoločným cieľom – doručiť zákazníkovi kvalitný produkt. Pre tieto dva tímy je najdôležitejšia vzájomná komunikácia. Keď sa však pre každý projekt najímajú noví tester, ktorí nemajú s vývojármi žiadny vzťah a sú v úplne neznámom kolektíve, je táto komunikácia veľmi formálna a neefektívna. Preto je dôležité mať statický tím testerov, rovnako ako to býva v prípade vývojárov. Dobré vzťahy a kolegiálna majú za následok výrazné zlepšenie spolupráce. Na dosiahnutie čo najlepšej komunikácie medzi tímom vývojárov a testerov je tiež dôležité zvoliť vedúce osobnosti oboch tímov, ktoré budú koordinovať ostatných členov tímu, budú mať prehľad o aktuálnom dianí v tíme a komunikovať s okolitým svetom v mene celého tímu.*

Kľúčové slová: *testovanie, vývoj, vzťahy, spolupráca, komunikácia, zodpovednosť*

Úvod

Softvér je produkt ako každý iný – je to výsledok snahy viacerých ľudí. V prípade softvéru to zahŕňa softvérových architektov, inžinierov, analytikov alebo vývojárov, pričom celý proces vývoja môže trvať aj roky. Cieľ tejto snahy je jasný – zisk. Ten je možné dosiahnuť (z dlhodobého hľadiska) len ak je zákazník spokojný. Zákazník je spokojný, ak je produkt,

ktorý si objednal/kúpil kvalitný. Inými slovami – cieľom celého vývoja softvéru je doručiť zákazníkovi kvalitný softvér. Čo je to vlastne tá kvalita a ako sa dá dosiahnuť?

Kvalita výrobku označuje, do akej miery daný výrobok spĺňa požiadavky zákazníka. V svete softvéru má na kvalitu vplyv viac faktorov. Jedným z najdôležitejších je testovanie softvéru. Je veľmi dôležité zvoliť vhodnú metódu testovania pre daný produkt, určiť kto, čo a kedy bude testovať. No je tu ešte jeden faktor, o ktorom sa príliš nediskutuje – komunikácia medzi vývojármi a testovacím tímom. Nevhodná, zle zorganizovaná komunikácia medzi týmito dvoma táborami môže mať v najlepšom prípade za následok výrazne pomalší vývoj/opravu zistených chýb, v najhoršom až zníženú kvalitu výsledného produktu.

Vzťahy, komunikácia a iné „drobnosti“

Dôležitosť dobrých vzťahov v tíme

Ak sa v kontexte softvérového vývoja spomenie komunikácia, každý má na mysli komunikáciu so zákazníkom, prípadne komunikáciu medzi členmi programátorského tímu. No z vlastnej skúsenosti viem, že rovnako dôležitá je aj komunikácia medzi vývojárskym a testerským tímom.

Predstava vymieňania informácií medzi týmito skupinami zahŕňa najmä zaznamenanie nájdenej chyby do systému na zaznamenávanie chýb zo strany testera a následné opravenie chyby zo strany vývojára. Prvá veľká chyba takéhoto „strojového“ prístupu je neosobnosť. Vývoj softvéru je práca ako každá iná – čím lepšie vzťahy na pracovisku a väčšia súdržnosť, tým väčšia spokojnosť pracovníkov, tým lepšia morálka tímu a najmä lepšie pracovné výsledky [3]. Priateľský vzťah medzi testerom a vývojárom je dôležitejší než akékoľvek iné vzťahy. Je to spôsobené už samotnou „nepriateľskou“ podstatou ich prác [4]. Jeden hľadá chyby vo výtvore toho druhého. No ak je ich vzťah na priateľskej úrovni, akékoľvek nepriateľstvo odpadá a jeden druhého berú viac ako partnera s rovnakým cieľom než tvorca problémov. Vývojár sa potom nebojí priamo poukázať na nejakú vlastnosť, ktorú treba pretestovať (čím sa dá ušetriť drahocenný čas v budúcnosti – chyba vyjde skôr či neskôr najavo) a tester dokáže popísať chybu spôsobom, ktorý je pre vývojára ľahko pochopiteľný – aj za cenu zníženej formálnej stránky opisu chyby. Taktiež s vývojárom zdieľa aj svoje testovacie taktiky, čo vývojárovi umožňuje rýchlejšie pochopenie chyby [4]. Aj preto je veľmi dôležité, aby mala firma stály testerský tím. V dnešnej dobe je totiž veľmi populárny prístup, pri ktorom sa na každý nový projekt volajú noví testeri [5]. Toto riešenie je samozrejme najjednoduchšie/najlacnejšie a je výsledkom podcenenia dôležitosti zohratosti medzi vývojármi a testerami.

Veľmi živo si spomínam na príchod nových testerov do našej firmy. Väčšia časť aplikácie bola naimplementovaná a otestovaná bola zatiaľ iba vývojármi. Preto bolo vhodné začať fázu testovania. Zatiaľ čo vývojársky tím mal za sebou už aj *team-building* stretnutia, testeri nepoznali ani jeden druhého. Aj napriek tomu, že mali skúsenosti s testovaním, stále to boli iba vystrašení vysokoškolskí študenti v novom prostredí. Spolupráca bola veľmi profesionálna – minimum osobnej komunikácie, každá chyba zapísaná podľa pravidiel. No identifikovanie a opravovanie chýb išlo veľmi pomalým

tempom. Keď mal vývojár nejasnosti v nahlásenej chybe, radšej zvolil komunikáciu cez nástroj na zaznamenávanie chýb ako osobný kontakt. Tento prístup dodržiava vnútro firemné osnovy, no je primárne určený pre veľké firmy, kde veľakrát nie je osobný styk ani možný. No naša firma je pomerne malá. Jedna z výhod malých firiem je flexibilita spôsobená malým počtom ľudí, zainteresovaných do takmer všetkých oblastí vývoja. A práve táto výhoda bola vytlačená nedostatkom komunikácie spôsobenej slabými vzťahmi. Všetko sa zlepšilo už v priebehu niekoľkých týždňov. Komunikácia bola menej strojená, uvoľnenejšia. Testeri už nemali strach poukázať aj na tie najmenšie nedostatky, ktoré neboli explicitne obsiahnuté v testovacích scenároch - vedeli že to vývojári nezoberú v zlom. Záznamy o chybách obsahovali menej formálne poznámky a aj interné pomenovania jednotlivých procesov a komponentov aplikácie. Pre vývojára to znamenalo okamžité pochopenie chyby. Veľakrát sa dokonca stalo, že tie najmenšie chyby ohlásili testerí priamo vývojárovi (ešte pred zaznamenaním do nástroja), ktorý ich okamžite dokázal opraviť. Výsledkom tohto priameho opravovania chýb, založenom na úzkej spolupráci a priamej diskusii o chybách bolo výrazné zefektívnenie celého procesu debugovania. Navyše, po istom čase začali testerí pomerne dobre chápať štruktúru aplikácie, lebo sa nebáli vývojárov pýtať na detaily. Tých zdatnejších zaujímali často aj implementačné detaily. Dôsledkom bolo, že veľakrát cielene odbočili od testovacieho scenára, lebo tušili možný výskyt chyby - ich podozrenie sa v niektorých prípadoch ukázalo ako opodstatnené, čím sa ušetrilo veľa času a nákladov v budúcnosti (čím neskôr sa chyba odhalí, tým je jej oprava náročnejšia a drahšia [6]).

Správne rozdelenie moci

Aj napriek výraznému zlepšeniu komunikácie sa čoskoro objavil ďalší problém. Ako som už spomínal, testerí sa často berú len ako spotrebné zboží - zavolajú sa keď je to potrebné, testujú dokedy je čo testovať a hneď na to ich poslanie vo firme končí. Podobne to prebiehalo aj u nás. V testerskom tíme nebola vytvorená žiadna hierarchia. Rozdelenie zodpovednosti bolo realizované len pridelením istého počtu scenárov na testovanie každému testerovi. Počet testerov postupne vzrástol na 6 členov a taktiež naplno začala fáza opravovania chýb. Tá zahŕňa aj pridelenie opravenej chyby na pretestovanie jednotlivým testerom prostredníctvom systému na správu chýb. Toto pridelenie mal na starosti samotný vývojár, ktorý po oprave chyby zmenil jej stav v systéme na opravenú. Vývojári ale nemali prehľad, ktorí testerí sú k dispozícii, či momentálne nepracujú na niečom zložitejšom/dôležitejšom a veľakrát sa muselo zbytočne dlho čakať na pretestovanie, kvôli prideleniu požiadavky nesprávnemu testerovi.

To všetko bolo výsledkom podcenenia situácie, spôsobeného nedostatkom skúsenosti manažmentu s väčšími projektmi. Podobné problémy sa riešia jasne definovanými rolami v rámci tímov [1]. Skupina ľudí s rovnakým cieľom a schopnosťami, vďaka ktorým je možné tento cieľ dosiahnuť sa nazýva tím [2]. Pre každý tím je veľmi dôležité mať člena, ktorý bude mať za úlohu koordinovať ostatných členov tímu. Táto osoba sa nazýva líder tímu (angl. team leader). Líder tímu musí mať neustále prehľad o aktuálne vykonávaných činnostiach v jeho tíme, má za úlohu prideľovať úlohy a najmä má na starosti komunikáciu s okolitým svetom [7]. Vo vývojárskom tíme bol líder tímu od samého počiatku implementácie, nakoľko sa implementácii po dokončení modelovania

4 Adam Pomothy

aplikácie venovala najväčšia pozornosť. No v tíme testerov sa na vedúcu osobu vôbec nemyslelo – pôvodne mal všetko zvládať projektový manažér. Čoskoro sa ale ukázalo, že projektový manažér nemá dostatočne podrobný prehľad o dianí v testerskom tíme.

Našťastie sa tomuto nedostatku v našom testerskom tíme venovala pozornosť dostatočne skoro. Manažment určil jedného z testerov za lídra tímu. Následne vývojári všetky otázky ohľadom nahlásených chýb adresovali práve jemu. V systéme na manažovanie chýb sa po opravení chyby pridelila požiadavka na pretestovanie opäť lídrovi testerského tímu. Ten mal najlepší prehľad o aktuálnej vyťažnosti testerov a taktiež vedel priradiť jednotlivým hláseniam o chybách a opravených chybách správnu prioritu. Vývojárom sa výrazne zjednodušila práca s hlásením požiadaviek na pretestovanie a svoj čas mohli venovať dôležitejším veciam.

Záver

Na vlastnej koži som zažil dôsledky podcenenia dôležitosti testerov vo vývoji softvéru. Testerí prispievajú k výslednej kvalite produktu minimálne rovnakým dielom ako samotní vývojári. Je dôležité, aby vývojári a testerí tvorili jeden celok s rovnakým cieľom a nepozerali sa navzájom na seba ako nepriatelia (aj keď podstata práce testerov je hľadať chyby vývojárov). Na to je dôležité mať utužené vzťahy medzi týmito dvoma tábormi. Jedným zo spôsobov ako dosiahnuť tento stav, je mať statický tím testerov a nemeniť ich pri každom projekte (čo je v súčasnosti v móde).

Ďalším dôležitým faktorom je správne rozdelenie zodpovednosti v tíme testerov. Ako každý iný tím, aj tím testerov potrebuje osobu, ktorá bude koordinovať činnosť ostatných členov a bude mať na starosti komunikáciu s okolitým svetom (v tomto prípade najmä s vývojárskym tímom). To umožní výrazne zefektívniť spoluprácu týchto dvoch skupín a zefektívni celý proces opravovania chýb. Takéto rozdelenie moci je úplne bežné pre tímy vývojárov, ale v prípade jednorazových testerov sa na dôležitosť podobnej zodpovednostnej hierarchie často zabúda.

Použitá literatúra

1. Addison, T., Vallabh, S.: Controlling software project risks: an empirical study of methods used by experienced project managers. In: *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*. Republic of South Africa, 2002.
2. DuBrin, A.J.: *Essentials of Management*. 6th ed. Peterborough, Ontario, Thomson South-Western, 2003.
3. Patil, T.: *What is the Best Way to Make Developer and QA Relationship Healthy?*, 2010.
4. Shammi, M.: *Relationship between a developer and a tester/QA*. 2008.
5. Soundararajan, P.: *What is the Best Way to Make Developer and QA Relationship Healthy?*. 2010.

6. Turbyfill , C.: Risk-Based Metrics for Software System Design, Development, and Test. *IEEE-USA Today's Engineer*, 2011.
7. Yujian Y., Xuelin N., Xiao X.: Three Level Competencies of Entrepreneurial Team: Team Leader, Member Level and Team Level. In *IEEE International Conference on Management of Innovation and Technology*, 2006.

Annotation

Everyone makes mistakes...that's what testers are for

Nowadays, testers are usually considered to be temporary one-shot employees. They are given only test scripts and rules, how to report found errors and they are supposed to do their jobs. However, testers are equivalent to a development team. They have the same goal – to deliver a high quality product to a customer. The communication is crucial for these two teams. However, if new testers are hired for every project, they have no relationship to developers and they are also in brand new environment. As a result, their communication is too formal and inefficient. That is the reason, why it is important to have a static team of testers as it is in case of developers. Good relationships or friendships result in much better cooperation. To achieve as effective communication as possible, it is also important to choose team leaders of both teams (it is common in case of developers, but not in case of testers). Team leaders coordinate other members of team, they are aware of current tasks and they communicate with external world in the name of whole team.