

ZNAMENÁ OPEN-SOURCE NIŽŠIU KVALITU?

„Pri dostatočnom počte očí sú všetky chyby plytké.“

Eric Raymond

Tomáš Caban

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

Abstrakt. *Open-source softvér zaznamenal obrovský úspech a dnes sa s ním stretávame takmer neustále. Otvorený softvér sa rozšíril aj do komerčnej sféry, keďže si spoločnosti uvedomili jeho nesporné výhody. Stále mu je však vytýkaná nedostatočná kvalita a spoľahlivosť, pričom tieto názory sú často argumentované nevyužitím klasických metód zaručenia kvality ako pri tradičnom vývoji softvéru. V tejto eseji sa zaoberám kvalitou open-source softvéru, metódami a prístupmi, ktorými sa dosahuje. Taktiež sa venujem otázke porovnania kvality open-source a komerčného softvéru a snažím sa odpovedať na otázku, či má open-source softvér nižšiu kvalitu ako komerčný softvér. V závere sa snažím zamyslieť nad možnosťami využitia metód, prostriedkov a techník open-source vývoja pri komerčnom vývoji za účelom dosiahnutia vyššej kvality.*

Kľúčové slová: *kvalita, open-source*

Úvod

Jedným z dôležitých procesov pri vývoji softvéru je manažment kvality s cieľom zabezpečiť kvalitný softvér. Kvalita softvéru je jednou z najdôležitejších požiadaviek na softvér a často je hlavným faktorom úspechu alebo neúspechu projektu.

Open-source softvér zaznamenal obrovský úspech od polovice 90-tych rokov. Vývoj softvéru s otvoreným kódom komunitami vývojárov a používateľov má mnohé výhody a

uvedomujú si to aj spoločnosti, ktoré často využívajú open-source softvér vo svojich vlastných komerčných projektoch. Open-source softvéru sú však často vyčítané nedostatočná kvalita a spoľahlivosť, keďže nevyužívajú štandardné techniky a postupy na zaručenie kvality ako pri tradičnom vývoji softvéru.

Proces vývoja open-source softvéru však obsahuje procesy a prístupy, ktoré slúžia na dosiahnutie kvality, a ktorým sa v tejto eseji budem venovať, aby som tak ukázal, že kvalita v open-source projektoch naozaj existuje. Budem sa tiež snažiť porovnať dosahovanie kvality v open-source projektoch a pri klasickom vývoji a nakoniec sa zamyslím ako by niektoré postupy a metódy z open-source vývoja softvéru mohli byť použité pri komerčnom softvéri.

Kvalitný open-source softvér

Open-source softvér je stále častejšie využívaný nie len komunitami vývojárov a používateľov, ale stále častejšie aj spoločnosťami, ktoré open-source softvér využívajú vo svojich projektoch alebo sa dokonca podieľajú na vývoji otvoreného softvéru. Stále však pretrvávajú obavy z nedostatočnej kvality open-source softvéru. Ako hlavné argumenty pre tieto obavy sa spomínajú chýbajúci centrálny manažment projektu, veľké množstvo vývojárov bez osobnej zodpovednosti, slabý návrh softvéru, neštruktúrovaný proces vývoja či nedostatočná dokumentácia. Musí to však znamenať, že open-source softvér je nekvalitný?

Pravdou je, že proces vývoja open-source softvéru umožňuje vytvorenie kvalitného výsledného produktu. Proces vývoja open-source softvéru zahŕňa aj techniky zaručenia kvality, ktoré sú síce odlišné od klasických techník využívaných pri vývoji komerčného softvéru, avšak podľa väčšiny výskumov o nič menej účinné s cieľom získania kvalitného softvéru. Otázkou kvality otvoreného softvéru sa zaoberalo viacero výskumov [4] a najčastejším výsledkom bolo, že kvalita otvoreného softvéru nebola výrazne odlišná v porovnaní s komerčným softvérom.

Ako sa teda proces vývoja otvoreného softvéru vysporiadáva s kvalitou bez využitia tradičných metód vývoja softvéru? Vývoj open-source softvéru sa spolieha na niekoľko kľúčových oblastí: udržateľné komunity, modularitu softvéru, manažment projektu a manažment procesu testovania. Je dôležité, aby účastníci procesu vývoja otvoreného softvéru dobre rozumeli týmto oblastiam a ich vzájomným závislostiam.

Udržateľná komunita

Vysoko kvalitný otvorený softvér sa veľmi spolieha na veľkú komunitu schopnú rýchlo vyvíjať program, efektívne ladiť a vytvárať nové funkcie. Zo štúdií [4] vyplýva, že získanie veľkej udržateľnej komunity by mal byť prvým a hlavným cieľom open-source projektu a zároveň je to jeden z kľúčových faktorov úspechu.

Komunita by mala byť rozdelená podľa úloh. Najčastejšie sa používa tzv. cibuľový model [4], ktorý tvoria malé množstvo jadrových vývojárov a rastúci počet prispievajúcich vývojárov, referentov chýb a používateľov. Tieto jednotlivé vrstvy by nemali byť uzavreté, malo by byť možné pre členov komunity posun smerom k jadru ako odmena a uznanie za prínos projektu.

Rozdiel medzi komerčným a open-source projektom je vo veľkosti vývojových tímov ale aj motivácii vývojárov a používateľov.

Veľmi dôležité je tiež vnímanie chýb používateľmi. Pri komerčnom softvéri používatelia očakávajú bezchybnosť a často nie sú ochotní tolerovať či nahlasovať chyby. Veď kto z nás už niekedy z frustrácie nevyvolal draho zaplatený softvér? Naopak pri open-source softvéri sú používatelia ochotní prispievať k opravám chýb, nakoľko sú si vedomí, že používajú bezplatný softvér, ktorý vytvárajú ľudia bez vidiny zisku. Výhodou je tiež väčšie množstvo používateľov pri väčšine open-source projektov.

Rozsiahla komunita je určite výhodou oproti obmedzeným počtom vývojárov a používateľov pri klasickom vývoji. Kritici často tvrdia, že veľké množstvo vývojárov produkuje nekvalitný kód na rozdiel od komerčného softvéru, kde sa prísne dodržiavajú definované postupy. Bolo vykonaných niekoľko výskumov [2], ktoré sa zaoberali meraním kvality kódu projektov s veľkým množstvom prispievateľov a potvrdilo sa, že aj veľké komunity dokážu produkovať kvalitný kód.

Modulárnosť softvéru

Modulárnosť softvéru predstavuje návrh softvéru, ktorý rozdeľuje funkcionality systému do niekoľkých nezávislých, zameniteľných častí (modulov). Pre každého vývojára je lákavé rozšíriť existujúci softvér samostatným modulom bez nutnosti zmeniť alebo porozumieť jadru systému. Znižuje to tiež riziko zavedenia nových chýb do ostatných modulov systému. Existujú tiež štúdie vývoja existujúcich open-source projektov [1], ktoré potvrdzujú, že modulárnosť softvéru zvyšuje záujem vývojárov pridať sa ku komunitě otvoreného projektu.

Ďalšie štúdie Apache a Mozilla projektov [1], ale aj ďalších 100 open-source projektov napísaných v jazyku C potvrdili, že vysoká modularita softvéru má pozitívny vplyv na kvalitu softvéru vyjadrenú hustotou chýb.

Manažment projektu

Dve oblasti, v ktorých sa vývoj open-source softvéru výrazne odlišuje od komerčného softvéru, a ktoré zároveň ovplyvňujú kvalitu, sú revízia seberovnými („peer review“) a manažment ľudí.

Revízia kódu nie je ničím novým a pri komerčnom vývoji je používaná už desiatky rokov. Rozdiel je v tom, že kým v open-source vývoji je revízia kódu kontinuálny proces, pri tradičnom vývoji je to jedno-, pri najlepšom dvojkrokový proces [1]. Ďalším významným rozdielom je počet ľudí, ktorí sa na revíziách zúčastňujú. Otvorené vývojové komunity majú k dispozícii omnoho viac ľudských zdrojov. Tento fakt však nepovažujú všetci za výhodu a existujú názory, že väčšie množstvo kontrolórov kódu nemusí byť bezpodmienečne efektívnejšie [3]. Revízia kódu sa v open-source projektoch vykonáva pred oficiálnymi vydaniami, čím sa môže dosiahnuť odhalenie chýb pomerne v skorej fáze životného cyklu softvéru, čo je často označované ako jeden z hlavných nedostatkov open-source vývoja.

Vytvorenie efektívneho prostredia a kultúry je takmer rovnako dôležité ako návrh systému. Proces open-source vývoja by mal aktívne nabádať ku kreativite a inováciám,

mal by byť viac zameraný na ľudí, ľudia by sa mali dobrovoľne pridávať k tímu a ponúkať sa na prácu na konkrétnej funkcionalite. Komunita open-source projektu by mala mať vytvorené procesy pre hlasovania za funkcie a účinné spôsoby komunikácie.

Komunita by mala využívať čo možno najviac komunikačných kanálov a efektívne využívať ich možnosti, aby oslovila čo najväčší počet členov komunity ale aj pritiahla nových vývojárov, používateľov či testerov. Okrem pre open-source komunitu takmer povinných komunikačných kanálov ako je mailing list a fórum je vhodné zväžiť aj ďalšie možnosti ako wiki či rôzne sociálne siete, Zároveň je však dôležité používať len také komunikačné kanály, ktoré budú aktívne využívané, v opačnom prípade by sa mohla komunita javiť ako nečinná a nových členov by lákala len veľmi ťažko. Niekedy sa treba riadiť heslom „Menej je niekedy viac“ a zvoliť naozaj len také kanály, ktoré budú živé. Dôležité je pri tom načúvať aj potrebám členov komunity. Ak napríklad má väčší počet členov záujem o fórum v inom jazyku, malo by sa im vyhovieť. Dôležité pre dosiahnutie efektívnej komunikácie v komunite je nesnažiť sa kontrolovať alebo riadiť tok informácií v komunite.

V oblasti manažmentu projektu je tiež veľmi dôležitá motivácia vývojárov. V komerčných projektoch je to jednoduché, vývojári sú motivovaní hlavne finančne, rôznymi bonusmi či kariérnym postupom. Kritici často vyčítajú open-source projektom, že vývojári nie sú tak motivovaní a teda neprodukujú kvalitný softvér. Veď kto z nás by niečo robil poriadne, keď to je zadarmo, však? Pravdou však je, že aj vývojári v open-source projektoch majú silnú motiváciu, robia to najmä pre osobné uznanie a možné budúce výhody napríklad v zamestnaní. Pre manažment open-source projektov je dôležité, aby umožnili takéto uznanie vývojárom, dali im možnosť uviesť svoje mená pri softvéri, postupovať v hierarchii komunity nahor či poskytnú ďalšie možnosti osobného uznania vývojárom. Vývojári tak ostanú motivovaní a produkujú kvalitný softvér aj bez finančných odmien.

Proces testovania

Pri vývoji komerčného softvéru je testovanie pravdepodobne najdôležitejšou súčasťou procesu dosiahnutia kvality softvéru. Open-source proces vývoja používa množstvo techník na dosiahnutie vysokej kvality.

Zo štúdie 200 open-source projektov [1] vyplynulo, že skoré a časté vydávanie má za následok neustále zlepšovanie softvéru veľkým množstvom prispievajúcich vývojárov na základe spätnej väzby od používateľov.

Štúdia 200 open-source projektov tiež ukázala, že menej ako 20% vývojárov používa testovacie plány, iba 40 % projektov používa testovacie nástroje a menej ako 50 % systémov používa nástroje na získavanie pokrytia kódu.

Open-source proces vývoja zjavne nepoužíva štruktúrované testovacie metódy, napriek tomu dokáže vytvoriť vysoko kvalitný softvér. Open-source projekty sa často spoliehajú na testovanie používateľmi [3], v niektorých prípadoch sa jedná o jediný typ testovania softvéru. V takýchto prípadoch je veľmi dôležité mať dostatočnú komunitu používateľov schopných a ochotných testovať systém pre všetky cieľové platformy, reportovať chyby a spolupracovať na ich odstránení.

Open-source kvalita v komerčnom softvéri

Ako bolo už podotknuté v tejto eseji, open-source proces vývoja dokáže vytvárať kvalitný softvér, podľa väčšiny výskumov rovnako kvalitný ako komerčný softvér. Tento fakt by sa mohol využiť pri zvyšovaní kvality komerčného softvéru použitím niektorých metód a procesov open-source vývoja bez nutnosti značného zvýšenia nákladov vynaložených na proces zaistenia kvality. Ako teda využiť open-source kvalitu v komerčnom softvéri?

Najväčšia prednosť open-source vývoja pri dosahovaní kvality je určite veľká komunita používateľov ochotných testovať a pomáhať pri vývoji. Ako však donútiť používateľov drahého komerčného softvéru, od ktorého samozrejme čakajú dokonalosť, spolupracovať pri hľadaní a riešení chýb? Mnohí výrobcovia softvéru už poskytujú aj voľné verzie svojich produktov, ktoré majú určitým spôsobom obmedzené niektoré funkcie, jadro softvéru zostáva však rovnaké a získajú tak väčšie množstvo používateľov, ktorí testujú ich softvér. Títo používatelia by tiež mohli byť prístupnejší k spolupráci pri riešení chýb, keďže softvér používajú zadarmo. Je tu však stále určitý blok, keďže používateľ nemôže používať všetky možnosti softvéru a tak v mnohých prípadoch nie je ochotný spolupracovať. Ďalšou možnosťou by mohlo byť poskytnúť používateľom komerčného softvéru, ktorí sa aktívne zúčastnia na testovaní, napr. poslaním chyby, určité odmeny. Či už najaktívnejším poskytnúť zľavu na softvér, zúčastneným poskytnúť napr. nejaké rozšírenie softvéru, nové grafické rozhranie, jednoducho snažiť sa používateľov motivovať. Tak by mohli získať lepšiu spätnú väzbu o chybách v softvéri a zvýšiť tak jeho kvalitu.

Ďalším dôležitým nástrojom na dosiahnutie vysokej kvality open-source softvéru je častá revízia zdrojových kódov veľkým množstvom vývojárov. V komerčnom vývoji nie je možné využitie toľkých ľudských zdrojov, je tu ale možnosť zaviesť revíziu kódu do komerčného vývoja ako kontinuálny proces vo všetkých fázach vývoja na rozdiel od jedno až dvojkrokového procesu revízie, ktorý sa využíva najčastejšie.

Veľmi dôležitým faktorom pre vytvorenie kvalitného softvéru je aj kvalitný a efektívny manažment projektu. K tomu patrí vedenie a motivovanie vývojového tímu. V oblasti komerčného vývoja sú vývojári motivovaní najmä finančne či kariérnym postupom. Open-source ale ukázal, že vývojárov možno motivovať aj bez finančných prostriedkov. Preto by bolo zaujímavé motivovať vývojárov v komerčnom vývoji osobným uznaním za tvorbu softvéru a to napríklad uvedením mena vývojára pri softvéri či jeho jednotlivých častiach. Takto by za softvér dostávali zásluhy nie len firmy, ale aj jednotlivci, ktorí by tak mali motiváciu ukázať sa v čo najlepšom svetle. Veď kto už by len chcel byť podpísaný pod nekvalitným softvérom, ktorého chyby privádzajú používateľov do zúfalstva? Určitú nevýhodu možno vidieť v tom, že pri takomto kroku by sa firmy mohli obávať straty svojich zamestnancov, ktorých by na základe zverejnených prínosov k softvéru mohla k sebe chcieť pretiahnuť konkurencia.

Pri dlhšom zamyslení by bolo určite možné vymyslieť aj ďalšie spôsoby ako by v komerčnom vývoji mohli byť využité metódy, procesy a prostriedky open-source vývoja na dosiahnutie vyššej kvality softvéru. Jedno je však isté, open-source vývoj dokáže produkovať vysoko kvalitný softvér a komerčný vývoj by sa mal namiesto kritiky kvality open-source softvéru z neho poučiť.

Záver

Kvalita a spoľahlivosť open-source softvéru je neustále spochybňovaná pričom najčastejším argumentom je, že sa nevyužívajú klasické metódy zaručenia kvality softvéru. Cieľom tejto eseje bolo ukázať, že open-source vývoj dokáže produkovať kvalitný softvér aj bez klasických metód.

V prvej časti sa esej venovala jednotlivým oblastiam vývoja open-source softvéru a popisovala ako sa dosahuje kvalita v procese vývoja softvéru. Tiež tu boli spomenuté rozdiely oproti klasickému komerčnému vývoju. Spomenutých bolo aj niekoľko rozsiahlych výskumov, ktoré sa zaoberali porovnávaním kvality open-source a komerčného softvéru, z ktorých jasne vyplynulo, že open-source softvér je rovnako kvalitný ako komerčný.

V závere sa esej pokúsila načrtnúť ako by sa dali použiť niektoré metódy, prostriedky či procesy open-source vývoja pre zvýšenie kvality komerčného softvéru.

Použitá literatúra

1. Aberdour, Mark.: Achieving Quality in Open Source Software. *IEEE Software archive*, Vol. 24, January 2007.
2. Howard, E., Norick, B., Krohn, J., Welna, B., Izurieta, C.: Effects of the number of developers on code quality in open source software: a case study. *ACM-IEEE*, 2010.
3. Khanjani, A., Sulaiman, R.: The Process of Quality Assurance under Open Source Software Development. *IEEE Symposium on Computers and Informatics (ISCI)*, 2011.
4. Srinivasan, R., Ashutosh, P., Birendra, K. M., Hsihui C.: Open Source Versus Closed Source: Software Quality in Monopoly and Competitive Markets. *Systems, Man and Cybernetics, Part A: Systems and Humans*. November 2005.

Annotation

Does open-source mean less quality?

Open-source software has achieved great success and today we encounter open-source software almost constantly. Open-source software has also spread to commercial area as companies had noted its undisputed benefits. However, open-source software has always been accused of lack of quality and reliability. Most used argument for this is that open-source development process does not use traditional quality assurance techniques and methods used in commercial development. This essay reviews quality assurance process in open-source development, methods, processes and techniques which provide quality of open-source software. This paper also aims to compare quality of open-source and commercial software and tries to answer question if open-source software really has less quality than commercial software. In the end this essay tries to think of possibilities of using open-source quality assurance in commercial development.