

TESTOVANIE AKO KAŽDODENNÝ CHLIEB

Kto nič nerobí, ten nič nepokazí.

Miroslav Beno

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
Autor[zavináč]mail[.]com

Abstrakt. Softvérové inžinierstvo pozná niekoľko metód na zabezpečenie požadovanej kvality výstupu tvorby informačných systémov. Jednou z nich je testovanie, ktorého nevyhnutnosť a zásadnosť časom podnietili vznik rôznych metodík, prístupov a teórií. Tým sa stala problematika testovania komplexnejšia a zároveň jej dobré zvládnutie náročnejšie. Dnes je už pri stredne veľkých projektoch takmer vylúčené, aby mali testovanie na starosti iba samotní programátori a tvorcovia systému. Informačné systémy bývajú popri tom často podrobené verifikácii špecializovaným tímom zameraným na nachádzanie chýb a nedostatkov a následné vytváranie správ o nich. To znamená, že existujú profesionálne zameraní odborníci práve na túto disciplínu - tester, ktorí musia spĺňať isté predpoklady pre úspešné realizovanie sa v tejto oblasti. Esej sa zaoberá testovaním z perspektívy profesijnej špecializácie, pričom polemizuje o opodstatnení jej existencie, a zároveň rozoberá rôzne formy a pozície v rámci tejto činnosti.

Kľúčové slová: testovanie, tester, profesia, kvalita, softvér, testovací tím, outsourcing

Kvalita, testovanie, teória

Človek nie je tvor dokonalý a mýliť sa je ľudské. Ostatne aj sám Albert Einstein povedal: „Každý, kto nikdy neurobil žiadnu chybu, nikdy nevykúšal nič nové.“ Toto obzvlášť platí pri tak produktívnej a prudko sa rozvíjajúcej oblasti, akou je softvérové inžinierstvo a vývoj softvérových systémov. Počas desaťročí nasadzovania softvérových riešení do prevádzky

sa ukázalo, že chyby a nepripravenosť produktov na neštandardné situácie bývajú často podceňované. Taktiež vyplynuli nie príliš pozitívne dôsledky tohto prístupu, ako napríklad to, že opravovanie nedostatkov v už široko používanom produkte je finančne náročnejšie, a preto je potrebné sústrediť sa na kvalitu produktu hlavne počas jeho vývoja. Overovanie kvality testovaním, za účelom identifikovania potenciálnych problémov, sa postupne stalo podstatnou a nevyhnutnou súčasťou vývojového cyklu softvéru.

Najskôr sa začala verifikovať základná funkcionálna stránka softvéru v podobe testovania zdrojového kódu, následne sa procesy testovania zovšeobecňovali na rôzne ďalšie oblasti. V súčasnosti sa v rámci týchto procesov vykonáva napríklad overovanie správania sa systému pod záťažou, sofistikovanými metódami sa zabezpečuje užívateľská prívetivosť systému, dôraz je kladený aj na testovanie bezpečnosti systémov. Okrem týchto takzvaných dynamických spôsobov testovania existujú aj techniky statického overovania, akými je napríklad akceptačné testovanie, kedy sa kontroluje dôslednosť splnenia zákazníkom zadanej špecifikácie systému. Samozrejme vyplynula prirodzená potreba všetky tieto procesy nejakým spôsobom formalizovať a kategorizovať. To podnietilo vznik mnohých metodík, pravidiel a teórií, ktoré umožňujú okrem iného merateľnosť a istú štandardizáciu testovacích postupov.

Doména testovania dozrela v natoľko komplexnú problematiku, že pritiahla aj pozornosť vedeckých odborníkov v informatike a stala sa predmetom mnohých štatistík a výskumov, analyzujúcich účinnosť a dopad testovania na softvérové projekty. Z tých najreprezentatívnejších, ponúkajúcich charakterizujúce pohľady na testovanie ako celok, zaujmú niektoré fakty a závery. Napríklad rôzne etapy testovania sumárne môžu zabrať až 40% času a rozpočtu projektu [1]. Zároveň predstavuje súčasná efektívnosť testovania v priemere 85% odhalených chýb pred uvedením softvérového produktu. Riešenie tých zvyšných môže stáť v závislosti od toho, v ktorej fáze projektu bola chyba zavedená, až 25-násobne viac ako keby boli odhalené a opravené ihneď [2].

Testovanie ako nočná mora?

Toto všetko je pekná teória, a myslím, že základnú opodstatnenosť vykonávania a manažovania procesov zabezpečujúcich kvalitu v softvérovom projekte si uvedomuje každý vývojár aj bez väčšieho premýšľania. Avšak v praxi býva ku kvalite a testovaniu pristupované s istým dešpektom a nízkou motiváciou, často bývajú aktivity s nimi súvisiace podceňované a zanedbávané. Aj z mojich vlastných skúseností, napríklad pri práci v rámci Tímového projektu v relatívne menšom tíme, býva kvalita často až poslednou otváranou témou. Veď oslniť nespočetným množstvom funkcií a nápadov je určite ambicióznejšie a možno aj jednoduchšie, ako zaujať dobre odladeným systémom, čo súvisí aj s časom potrebným na ocenenie a uznanie týchto črt softvéru. Aj keď pri vývoji v akademickom prostredí je ešte bagatelizovanie kvality softvéru viac ospravedlniteľné ako v komerčnom sektore hroziacim vážnymi následkami.

Neexistujú ale akési primárne psychologické bariéry vývojára vysvetľujúce nie práve nadšený postoj k testovaniu? Tou evidentnou je to, že testovanie sa nezdá byť kreatívnou činnosťou. Nemôžeme sa obzrieť a povedať si: „Mám za sebou určité množstvo práce!“. Nie je viditeľný hmatateľný a konkrétny pokrok ako napríklad pri pridávaní funkčnosti. V určitých aspektoch testovanie postup práce nielen spomaľuje ale dokonca redukuje. Treba

sa vrátia k už zrealizovaným funkciám, ktoré nepracujú tak ako by mali a niekedy menia okrem implementácie aj ich návrh. Ďalším, možno o niečo viac podvedomým efektom je skutočnosť, že pri testovaní vývojár nachádza vlastné chyby, prichádza na nedostatky vo svojom myslení, alebo na povrchnosť pri vytváraní softvéru. Je zrejмый demotivujúci vplyv takejto činnosti, keďže všetci sa chceme radšej utvrdzovať v tom, akí sme dobrí, a nie v tom, že robíme chyby. Neoblúbenou vlastnosťou testovania je aj jeho nejasná ohraničenosť. Otestovať softvér kompletne na zamedzenie akejkoľvek chyby je prakticky vylúčené a nájdenie krehkej hranice určujúcej dostatočnú kvalitu, a teda termín ukončenia testovania, je často veľmi komplexnou otázkou.

A teraz si predstavme, že jestvujú ľudia, pre ktorých testovanie znamená profesijnú činnosť, oblasť, ktorej sa primárne venujú denno-denne. Základné predpoklady na existenciu testovania a všeobecného manažmentu kvality ako profesie už boli načrtnuté. Hlavným je to, že problematika testovania je v súčasnosti dostatočne komplexná a na jej úspešné uchopenie je často krát žiaduce dobré zvládnutie špecifických zručností, schopností a skúseností. Ďalším je jasná rentabilita efektívneho testovania, a teda tým pádom aj možnosť finančne ohodnotiť jeho dôsledné realizovanie. Ako argument potvrdzujúci tieto predpoklady možno jednoducho konštatovať fakt, že pozície profesionálnych testerov s rôznou mierou požadovanej odbornosti sú už v praxi etablované a pomerne presne vyprofilované.

Po predchádzajúcom vymenovaní niektorých špecifik testovania, a pribudnutí obáv z prílišnej repetitívnosti a monotónnosti vo vykonávanej práci, môže takáto predstava v prvom okamihu pôsobiť dosť odstrašujúco a neatraktívne. Iste, zanikne nám síce spomínaný efekt nachádzania vlastných chýb, ale tento môže byť nahradený ďalšou, pre mnohých ešte nepríjemnejšou myšlienkou „hrabania sa v cudzom kóde“. V skutočnosti sa samotní testeria s metódou bielej skrinky, ako sa nazýva takýto prístup, pri ktorom sa testujú interné štruktúry aplikácií a zároveň je potrebná orientácia v kóde, príliš do kontaktu nedostávajú. Stále je však prítomná nutnosť oboznamovať sa s nejakým novým, cudzím projektom a jeho procesmi, čo si vyžaduje značnú mieru koncentrácie a trpezlivosti. V súvislosti s tým mi nedá nespomenúť si na rozhovor s jedným špecialistom z IT brandže, ktorého zameranie nebolo konkrétne testovanie, ale analýza a hodnotenie možnosti začlenenia a prevzatia existujúcich menších projektov do komplexného integrovaného systému, v dôsledku rušenia neúspešných divízií IT spoločnosti. Z jeho rozprávania vyplynulo, že napríklad skúmanie cudzieho kódu možno vnímať aj ako spoznávanie myslenia a prístupu druhých, a časom pozorovať až akési archetypy softvérových vývojárov, charakterizujúce mieru ich odborných a povahových vlastností. Priznám sa, že v takomto kontexte som nad testovaním nerozmýšľal, možno preto, lebo som sa s ním nestretol v jeho rýdzej forme. Áno, pri vytváraní softvéru často používame už existujúce komponenty a riešenia iných, ale miera pochopenia sa vzťahuje iba na analýzu zdokumentovaných rozhraní.

S prácou testera sa teda možno stačí podobne iba bližšie zoznámiť bez predchádzajúcich predsudkov, na to, aby sme zistili aké sú jej pravé špecifiká a charakteristiky, dokázali takúto profesiu primerane oceniť, a prípadne zistili, že obnáša aj celkom zaujímavé postupy a netriviálne požiadavky.

Začiatky a základy testovania

Ku každej špecializácii v odbore vedie postupná cesta, ktorej začiatok môže byť motivovaný istými okolnosťami. Ideálnou predispozíciou na kariéru softvérového testera je oboznámenie sa so základmi tohto zamerania už počas štúdia softvérového inžinierstva. Významné postavenie testovania v rámci tohto odboru podmienilo napríklad aj vznik úspešného špecializovaného semestrálneho predmetu ponúkaného už 5 rokov prestížnou Utah Valley University [3]. Z jeho koncepcie môžeme odpozorovať primárne aktivity spojené s prácou začínajúceho testera a spoznať motivačné princípy takejto práce, ktoré študentom predkladá.

Kurz správne vychádza z predpokladu, že v období štúdia, a s eventuálnymi prvými skúsenosťami z praxe, študent pristupuje k softvéru hlavne z dobre známej pozície vývojára menších projektov. Preto je predmet koncipovaný v prvom rade na prezentovaní odlišností vo vnímaní softvéru z pohľadu testerov a vývojárov. Samozrejme cieľom je aj nadobudnutie skúseností vďaka aplikovaniu rôznych testovacích techník a výberu správnych metód testovania. Študenti vypracovávajú projekt v dvoch časovo rovnocenných fázach predmetu. V prvej majú za úlohu vyvinúť aplikáciu s menej zložitou, ale nie triviálnou funkčnosťou a následne vykonať jej otestovanie, zamerané najmä na zdokonalenie kvality zdrojového kódu. V druhej časti si študenti medzi sebou svoje projekty vymenia a aplikujú pokročilejšie testovacie techniky zaoberajúce sa viacerými aspektami softvéru. Zadaní je niekoľko, tak, aby bol na testovanie pridelený typ aplikácie odlišný od tej, ktorú študent vyvíjal. V rámci tejto fázy sa už pracuje s formálnejšími postupmi produkujúcimi rôzne množstvo testovacích artefaktov. Základným je takzvaný testovací prípad (anglicky test case), ktorý opisuje vstupy, podmienky, vykonané akcie alebo udalosti a očakávaný výsledok, správanie aplikácie. Potrebné je zvoliť širokú množinu testovacích prípadov v rôznej miere podrobnosti, validujúcu rôzne časti aplikácie. Táto množina sa následne po vyhodnotení, spolu s uvedenými reálnymi pozorovanými výsledkami, vracia späť k vývojárovi v súbore hlásení o chybách (defect/bug report), ktorý zasadzuje jednotlivé testovacie prípady do kontextu a ponúka aj ďalšie sumarizujúce závery obsiahlejšieho testovania. Fundamentálnym dokumentom, ktorý sa študenti učia vypracovať je aj testovací plán (test plan) definujúci a inicializujúci procesy testovania. Opisuje ciele, rozsah a konkrétne prístupy testovacieho úsilia a mal by podávať celkový obraz o tom, prečo a ako sa bude systém testovať.

Okrem teoretických a formálnych skúseností si študenti vďaka projektu odnášajú aj poznanie, že vývojár je často vývojom ovplyvnený natoľko, že nedokáže sám o sebe vykonať dostačujúce relevantné testovanie. Súvisí to s odlišnými stimulmi testovania – vývojár sa snaží testovaním dokázať, že softvér funguje a nemá chyby, pričom tester naopak vie, že softvér chyby obsahuje, a cieľom jeho aktivity je ich nájsť.

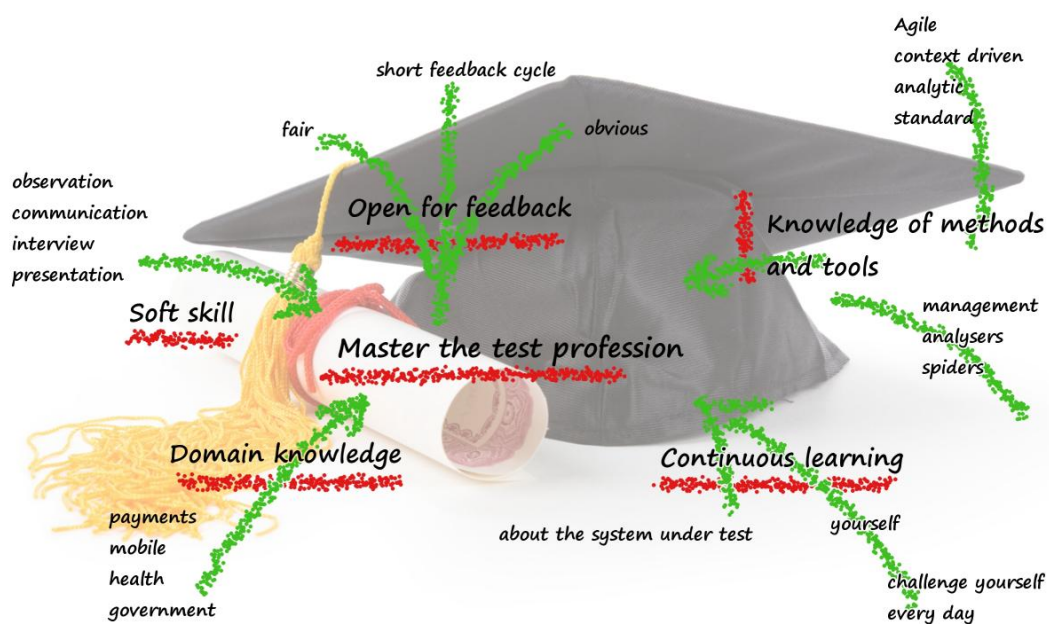
Je dôležité si uvedomiť, tak ako to zdôrazňujú aj autori kurzu, že ohodnotenie testera nezávisí na počte nájdených softvérových defektov, ale na kvalite jeho testovania – či boli vhodne zvolené testovacie metódy, správne odhadnuté kritické body systému a zrozumiteľne a dôkladne spracované výsledky testovania.

V súvislosti so vzdelávaním sa v testovaní, je zaujímavá skutočnosť, že táto konkrétna oblasť je v určitých smeroch štandardizovaná natoľko, že napríklad americké inštitúty ponúkajú možnosti certifikácie znalostí testovania. Uchádzač môže formou testu

preukázať svoje formálne a teoretické vedomosti a podporí tak profesionálnu aspiráciu na testovacieho špecialistu. Pravdou je, že takéto certifikáty nemôžu zaručiť reálnu kompetentnosť ich držiteľov a ich relevantnosť v praxi býva často vnímaná kontroverzne. Je totiž ťažké merať individuálnu produktivitu, praktické skúsenosti a ďalšie vlastnosti, ktoré, ako si ukážeme, často v značnej miere prispievajú ku kvalitnej práci a profesionalite testera.

Ideálny tester

Keď sa nad prácou testera zamyslím z môjho pohľadu, pod ideálnymi vlastnosťami a povahovými črtami vhodnými na vykonávanie testovania si predstavím napríklad schopnosť pozeráť sa na veci z rôznych uhlov a neštandardných perspektív. Dobrý tester by mal, podľa môjho názoru, byť vynaliezavý a prahnúť za kvalitou s dôrazom na details. Nevyhnutným základom je samozrejme zvládnutie formálnych zručností a teoretických techník testovania. V praxi sa ale ukazuje, že pôsobenie testera môže byť oveľa širšie ako sa zdá. Jeho efektívita sa rozširuje vplyvom ďalších osobnostných kvalít a zdokonaľovaním sa v príbuzných schopnostiach. O konkrétnych charakteristikách polemizujú mnohí odborníci v oblasti softvérového testovania na základe svojich dlhoročných skúseností. Skúmaním týchto charakteristík dokážeme preniknúť hlbšie do podstaty práce testera. Ilustračný štruktúrovaný prehľad jej hlavných elementov vidíme na Obr. 1.



Obr. 1. Povahové charakteristiky a schopnosti vyžadované pri práci testera (prevzaté z [4]).

Prvým z kritérií sú komunikačné, prezentačné a niekedy až diplomatické schopnosti. Tester by mal vedieť primeraným spôsobom vyjadrovať svoje názory a myšlienky v interakcii s inými ľuďmi. Kritická časť práce testera spočíva totiž v komunikácii s vývojármi, a ak si chce vybudovať určitú kredibilitu musí vytvárať názorné, zrozumiteľné a na druhej strane zároveň dostatočne technicky špecifické závery testovania. Je dobré vedieť odhadnúť ako bude poskytnutá správa o testovaní pôsobiť a rozmýšľať nad spôsobom ako sú konkrétne problémy opísané. Pokým sa totiž nezdvajú výstupy testovania vývojárovi užitočné, môže ich jednoducho zamietnuť a označiť za nerelevantné. Vhodné sú teda dostatočné jazykové, ale aj argumentačné dispozície, keďže tester musí často krát napríklad zdôvodňovať prečo je daný nález naozaj chybou a aké riziká pri používaní softvéru z neho vyplývajú. Tester by mal byť tým, ktorý udržiava základy asertívnej komunikácie v akejkoľvek situácii a prispeje tak k východisku prospešnému pre obe strany, aj projekt samotný.

Nasledujúci princíp by sa dal aplikovať aj všeobecnejšie na oblasť informačných technológií, ktorá sa neprestajne dynamicky rozvíja. Je ním schopnosť neustále sa vzdelávať a učiť sa, respektíve skúšať, nové veci. To sa netýka iba technologických noviniek. Testovanie môže byť často veľmi variabilnou disciplínou s individuálnym prístupom. Ako možno trochu nadnesene a agitačne napísal Rob Lambert vo svojom príspevku [5], dobrý tester by mal sústavne hľadať nové výzvy, aj čo sa týka svojich schopností, aj v rámci zaužívaných postupov a teórií. To znamená, že pri konkrétnych projektoch uplatní rôzne špecifické rozhodnutia ovplyvnené osobitosťami projektu a jeho postupným vývojom a nebude sa slepo pridržať všeobecných pravidiel a metodík. Jednoducho povedané, testovanie nemusí vychádzať zo stanovených dogiem a ak sa odchylením od nich dá prehĺbiť jeho výsledok, tester by mal túto možnosť využiť. Môže prichádzať s novými spôsobmi ako vytvárať návrh testovania, zvyšovať použiteľnosť výstupov alebo dokonca modifikovať za týmito účelmi samotný subjekt testovania.

S predchádzajúcim prístupom súvisia aj požiadavky na flexibilitu testera a schopnosť reagovať na spätnú väzbu. Spomínané kreatívne procesy v rámci testovania je vhodné konzultovať s ďalšími kolegami, či už je s nimi tester v priamom pracovnom vzťahu alebo nie, a reagovať na ich prípadné pripomienky. Dôležité je vyjadrenie svojho názoru a následné akceptovanie mienky ostatných. Tak ako pri každom „remesle“, aj tu môže byť kolaboratívne úsilie podporené vzájomnou otvorenou komunikáciou iba výhodou, a pomáha danému odboru ako celku napredovať a prinášať nové poznatky a lepšie riešenia.

Ruka v ruke s praktickými zručnosťami ide aj snaha o ich efektívne aplikovanie. Svoju činnosť tester v praxi realizuje prostredníctvom viacerých podporných nástrojov. Mal by si preto udržiavať súbor rozličných špecializovaných aplikácií a orientovať sa v konkrétnych aktivitách, pri ktorých jednotlivé nástroje asistujú. Môže ísť o podporu celkového manažmentu práce, štandardné vývojové prostredia, špecializované komunikačné nástroje alebo rôzne formy automatizovaného testovania.

Pre správnu analýzu a vyhodnocovania dopadu chýb je nevyhnutná znalosť domény projektu. Ak napríklad testuje systém realizujúci bankomatové služby, je dobré okrem technických špecifik poznať pridružené informácie ako tvar finančných transakcií, poplatkov, spôsob akým je prepojená sieť rôznych bánk a podobne. Pochopiteľne je nereálne, aby tester mal prehľad o vzťahoch a súvislostiach v rámci všetkých domén, s ktorými príde do styku. Preto sa musí vedieť správne v rozumnom čase zorientovať v

základných princípoch danej domény tak, aby vedel vyextrahovať jej rizikové oblasti a potenciálne problémy a zameral na ne testovanie.

Testovanie ako služba

Z predchádzajúceho opisu je zrejmé, že obsiahnutie týchto profesijných aspektov na potrebnej úrovni určite nie je triviálnou záležitosťou. Na ich zvládnutie potrebuje tester vhodné zázemie s príslušnými štruktúrami a náležitosťami, ktoré bude zamerané aj na podporu rozvíjania jeho schopností. V súčasnosti existujú firmy a organizácie, v ktorých portfóliu figuruje testovanie ako jedna z hlavných ponúkaných služieb. Je preto logickým smerovaním profesionálneho testera pôsobenie v jednej z takýchto spoločností, kde má vhodnejšiu pôdu na svoju realizáciu, ako keby vykonával svoje kompetencie len v rámci všeobecne zameraného softvérového tímu. Takýto spôsob práce bol umožnený postupnou modernizáciou a modularizáciou procesov vývoja softvéru, kedy sa dá testovanie realizovať externým špecializovaným subjektom v podobe takzvanej outsourcovanej služby. Nemá však takýto prístup aj negatíva, ktoré by možno potláčali opodstatnenosť profesionálneho testovania? Stručne sa môžeme pozrieť na to ako a ktoré úlohy je vhodné preniesť do externého prostredia, kde ich efektívne a na mieru vykonajú profesionáli, a ktoré z odborných metód pri tom takíto testerí používajú.

Argumentom za outsourcovanie testovania je zvýšená efektivita práce v profesionálnom prostredí. Štatistiky uvádzajú, že popredné organizácie vykazujú posun v odhaľovaní chýb pred vydaním softvérového produktu na viac ako 95% [2]. Ďalším benefitom je zrýchlenie vývoja softvéru, kedy testovanie môže prebiehať s ním simultánne a vývojový tím je odbremený od časovo náročných testovacích úkonov. V niektorých prípadoch sa tieto dokonca externalizujú spôsobom cez deň vývoj, a v noci – v inej krajine a časovom pásme – testovanie. Vedľajším, ale užitočným efektom je aj to, že pri samotnom formulovaní požiadaviek na testovanie sa často krát objavia nejasnosti a potenciálne problémy v návrhu systému, ktoré treba analyzovať, v dôsledku čoho sa zvýši jeho výsledná kvalita. V sumári sa všetky tieto aspekty dajú zjednotiť v hlavnej výhode profesionálneho testovania, ktorou je výhodnejšie investovanie finančných zdrojov v rámci produkcie softvéru.

Na to, aby bol tento zámer úspešne naplnený aj v praxi, je ale potrebné pristupovať k takémuto testovaniu vhodným spôsobom. V prvom rade musí mať projekt dostatočný rozsah, aby réžia spojená s outsourcovaným testovaním nebola väčšia ako výsledný úžitok. Ďalej sa posudzujú rôzne úrovne a vhodné formy pre takéto testovanie. Prehľadný rozpis vidíme na Tab. 1.

Z pohľadu úrovne sa testujú viac časti softvéru s vyššou mierou abstrakcie [6]. Napríklad verifikáciu konkrétnej funkčnosti častí kódu väčšinou zabezpečujú vývojári (anglicky unit testing), na rozdiel od testovania systému ako celku podľa špecifikácie (system testing), ktoré zahŕňa aj nefunkcionálne požiadavky. Z hľadiska typu testovania sú vhodné napríklad techniky regresného testovania (regression testing), ktoré je zamerané na odhaľovanie chýb vynárajúcich sa opakovane na kritických miestach po pridaní rôznej funkčnosti softvéru. Menej efektívne je externalizovať záťažové testy systémov (performance testing) alebo bezpečnostné testy. Pri pohľade na testovacie procesy je samozrejším základom služby esenciálne vykonávanie testov a podávanie ich

výstupov. Celkové manažovanie testovania by však malo ostať v rukách hlavného manažéra softvérového projektu a nie vedúceho tímu testerov.

Tab. 1. Vhodné formy testovania pre outsourcing [6].

Veľmi vhodné	Niekedy vhodné	Nevhodné
Úroveň testovania		
systémové testovanie	„unit“, integračné testovanie	akceptačné testovanie
Typ testovania		
regresné testovanie, testovanie funkcionality, lokalizácie, kompatibility	záťažové testovanie, testovanie použiteľnosti, bezpečnosti	
Aktivity testovania		
vykonávanie testov, hlásenia výstupov, automatické testovanie	plánovanie testovania, analýza a dizajn testovania (vyžaduje dobrú znalosť biznis domény)	manažment testovania

Pri outsourcovanom testovaní je základom úspešnej spolupráce dobrá komunikácia medzi klientom a testovacím tímom. Dôležitým predpokladom je aj to, aby výsledky testovania mohli byť zhodnotiteľné a tým pádom merateľné. Treba teda identifikovať kľúčové indikátory výkonnosti, načo sa používajú metriky ako testovaním pokrytý rozsah, aká časť testovania bola automatizovaná, a množstvo neobjavených chýb po skončení určitých testovacích fáz. Na testerov v takejto situácii je v dôsledku toho kladená aj istá miera administratívnej činnosti pri zvýšenom dokumentovaní svojej práce.

Tester - softvérový hrdina

Zámerom tejto eseje bolo ponúknuť špecifický pohľad na softvérové testovanie ako profesiu z viacerých stránok. V texte som sa snažil konfrontovať môj pohľad, ktorý zatiaľ nebol formovaný praktickými skúsenosťami s reálnym testovaním, s faktami a publikáciami z odbornej praxe. Motiváciou na rozobratie testovania z tejto perspektívy bolo to, že tak dôležitá súčasť softvérového inžinierstva, ktorá zásadným spôsobom ovplyvňuje výsledný produkt, je často na okraji záujmu budúcich absolventov tohto odboru z môjho okolia. Presvedčil som sa, že tieto postoje nie sú úplne opodstatnené a byť dobrým testerom je zložitejšie, ale aj zaujímavejšie, ako sa možno na prvý pohľad zdá.

Softvérové testovanie môže predstavovať pre človeka výzvu, keďže sa jedná komplexné povolanie, ktoré si vyžaduje okrem pokročilej schopnosti vnímať aj trpezlivosť a ďalšie spomínané vlastnosti. Zároveň sú teória a rôzne metódy a oblasti testovania dostatočne netriviálne na to, že môžem skonštatovať, že moje počiatočné obavy, či nie je testovanie monotónne a nezaujímavé, boli prehnané. Po spoznaní procesov, ktorými sa softvérový tester zaoberá si myslím, že táto práca býva často dosť nevďačná a podceňovaná. Odborníkov z tejto sféry je potrebné si vážiť a možno aj odhodlať sa a aspirovať na ich pozície.

Použitá literatúra

1. NIST. The economic impacts of inadequate infrastructure for software testing. 2002. [cit. 2010-10-15] Dostupné na internete: <<http://www.nist.gov/director/prog-ofc/report02-3.pdf>>.
2. Jones, Capers. *Software technology support center* [online]. 2008 [cit. 2010-10-17]. Measuring Defect Potentials and Defect Removal Efficiency. Dostupné z WWW: <<http://www.stsc.hill.af.mil/crosstalk/2008/06/0806Jones.html>>.
3. K Harrison, N. B. Teaching software testing from two viewpoints. 2010. J. Comput. Small Coll. 26, 2 (Dec. 2010), 55-62.
4. Prins, Andréas. *Software Testing and more* [online]. 2010 [cit. 2010-10-13]. Master the test profession. Dostupné z WWW: <<http://www.testingthefuture.net/2010/06/master-the-test-profession/>>.
5. Lambert, Rob. *The Social Tester* [online]. 2010 [cit. 2010-10-19]. Don't be a follower, be a tester. Dostupné z WWW: <<http://thesocialtester.posterous.com/dont-be-a-follower-be-a-tester>>.
6. Moser, Silvio. Outsourced Testing – Friend or Foe?. *Methods & Tools* [online]. 2008, roč. 16, č. 3, [cit. 2010-10-10]. Dostupný z WWW: <<http://www.methodsandtools.com/archive/archive.php?id=74>>.

Annotation

Testing as a daily bread

Software engineering recognizes several methods that are used in order to ensure that the output quality meets the requirements given by software systems. One of the methods is testing. Throughout time, due to the relevance and essentiality of this method, several different methodologies, approaches and theories were established. Therefore the issue of testing became more complex and hence more difficult to master. As a result, even in medium-sized projects, it cannot be performed by programmers and system developers only at the present time. Instead, software systems are subjected to specialized verification, which is focused on finding errors and weaknesses. The verification is performed by a specialized team, which outlines the results in reports. The professionals in the team are experts, who focus on the testing discipline only, and have to meet certain prerequisites in order to be qualified for the job. This essay deals with testing from their point of view. It discusses the justification of existence of such testing professions and explores various forms of testing and job opportunities in this area.