

MÁ ZMYSEL TESTOVAŤ SOFTVÉR MANUÁLNE?

„Včera to fungovalo, ale radšej to pretestuj!“

Filip Sucháč

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

Abstrakt. Každá softvérová firma chce, aby bol jej výsledný produkt kvalitný. Určitá časť procesov, ktoré sa zaoberajú dosahovaním kvality softvéru, sa nazýva testovanie. Prístupov k testovaniu je viacero. Niektoré firmy sa počas implementácie viac spoliehajú na automatické testy, alebo že si vývojári svoju prácu sami po sebe priebežne testujú. Vo viacerých firmách však existuje pozícia tester, obsadená hneď viacerými ľuďmi, ktorí vyvíjaný produkt dookola manuálne testujú a upozorňujú na prípadné chyby. V tejto eseji sa zaoberám výhodami a nevýhodami manuálneho testovania, tým aké vlastnosti by mal mať správny tester, ktorý testuje manuálne, a ktoré prístupy k testovaniu je dobré si zvoliť.

Kľúčové slová: kvalita softvéru, manuálne testovanie, prieskumné testovanie, chyby

Úvod

Nikto nie je dokonalý. Tento výrok platí pri akejkoľvek ľudskej činnosti a to aj pri vývoji softvéru. Napriek tomu sa na to aj v tejto oblasti často zabúda, čo môže mať nemalé následky. V priebehu vývoja môže vzniknúť množstvo nežiaducich chýb z rôznych dôvodov, od nepochopenia zadania až po nevedomý preklep v zdrojovom kóde. Testovanie je preto nutnosť, ktorej vynechanie vopred určuje projekt na neúspech.

V dnešnej modernej dobe, v ktorej sa všetko snažíme zautomatizovať, nie je ani táto dôležitá činnosť výnimkou tohto javu. Testovanie pomocou automatických testov často nahrádza alebo vytláča testovanie, ktoré je robené manuálne, teda priamym kontaktom

2 Filip Sucháč

človeka s produktom. Podľa môjho názoru však dôležitosť manuálneho testovania tak skoro nezmizne.

Manuálne vs. automatizované testovanie

Ako som už naznačil, manuálne testovanie je opozitum automatizovaného testovania. Kroky testovacích scenárov, ktoré vykonáva tester manuálne, pri automatizovanom testovaní vykonáva program alebo skript. Oproti manuálnemu testovaniu to má množstvo výhod ako je napr. väčšia rýchlosť testovania. Navyše tým, že to za človeka vykonáva stroj, je možné spustiť testy nad veľkým objemom dát cez noc, čo môže výrazne ušetriť čas aj peniaze. Aby sa však dal test spustiť, je potrebné ho vytvoriť. Vytvorenie testu v prvom rade zaberie nejaký čas, čo však nie je zlé, ak zoberieme do úvahy čas, ktorý dokáže takýto test oproti manuálnemu otestovaniu ušetriť. Ak by ale otestovať konkrétnu vec ručne zabralo menej času, potom by otestovanie tejto veci zautomatizované nebolo až také výhodné. Tento ručný test by však musel byť potrebný na použitie čo najmenej krát, pretože jeho opakovaním by strávený čas narastal a mohol by prevýšiť aj čas, za ktorý by mohol byť vytvorený automatický test.

Ďalším dôležitým faktorom v diskusii medzi manuálnym a automatizovaným testovaním je fakt, že pri automatizovanom je potrebné testy udržiavať. Zdrojový kód, ktorý tieto testy testujú, sa v priebehu projektu často mení. Menia sa požiadavky alebo sa kód refakturuje, teda mení jeho štruktúra. Preto si aj testy často vyžadujú zmeny. Navyše, ak sú testami programy alebo skripty pozostávajúce zo zdrojového kódu, na to aby boli schopné byť neustále menené je potrebné, aby boli prehľadné a „čisté“[4]. Preto po zmene ich funkcionality sa treba venovať aj ich refaktoringu. Výhodou v manuálnom testovaní je, že sa ho tento problém netýka a teda nepotrebuje na neho vynakladať toľko času. V prípade zmeny funkcionality produktu sa namiesto zmeny testov manuálny tester len oboznámi s novými krokmi testovacích scenárov danej funkcionality.

Scenár vs. prieskum

Pri testovaní chcem spomenúť ešte dva spôsoby testovania:

- testovanie podľa testovacích scenárov, pri ktorom musí tester vykonať vopred určené kroky v predpísanom poradí
- prieskumné testovanie, kde tester nepostupuje podľa scenára, rozhoduje sa sám aké kroky zvolí

Prvý prístup, teda podľa testovacích scenárov, môže byť využívaný manuálnym aj zautomatizovaným testovaním. Prieskumné testovanie je zatiaľ možné robiť len manuálne, nakoľko v ňom hrá hlavnú rolu rozhodovanie sa človeka testera. To, že tester nemusí postupovať podľa scenára neznamená, že sa môže flákať a postupovať hocijakým spôsobom. Bez scenára je to ešte náročnejšie, pretože by sa mal o vyvíjanom softvéri učiť, zisťovať rôzne podrobnosti a uvažovať nad tým, čo by mal spraviť aby odhalil chyby. Phillip G. Armour[1] tvrdí, že testovanie je pravdepodobne viac než akákoľvek iná aktivita vo vývoji softvéru o objavovaní. Tester sa snažia prísť na to, či nevedia, že niečo nevedia. Toto sa podľa mňa robí najlepšie manuálne, keď tester spoznáva testovaný softvér priamo

pod svojimi rukami. Vďaka tejto priamej interakcii a jeho znalostiam o softvéri prichádza na nové kroky, ktoré hneď aj skúša. Ak odhalí chybu, okolnosti, pri ktorých ju odhalil môžu poslúžiť ako základ pre ďalší test, pri ktorom predpokladá ďalšiu chybovosť softvéru.

Aký by mal/nemal byť dobrý tester

Súhlasím s tým, čo napísal Elliot Hamai[3], že tester nemusí mať nevyhnutne programovacie znalosti, hoci môže byť výhodnejšie ak ich má. Dôležitejšie je ale logické myslenie a systematický prístup k riešeniu problémov. Mal by mať otvorenú myseľ pre „hranie“ sa s problémami. Fantázia tiež môže byť pri testovaní prínosom. Dôležitá je tiež schopnosť učiť sa, aby vedel nadobúdať znalosti o testovanom softvéri a používať ich pri práci s ním, ako aj v budúcnosti pri ďalších projektoch.

Skúsenosti viacerých ľudí svedčia o tom, že vývojári nie sú úplne vhodní na testovanie produktu[5], obzvlášť pri manuálnom testovaní. Sú to proste programátori, ktorí produkt priamo vyvíjajú a nie tester. Rozmýšľajú a správajú sa inak. Napríklad, vývojári prechovávajú ku kódu „rodičovské pocity“, pretože ho sami vytvorili. Sú týmto faktom natoľko ovplyvnení, že im to bráni vidieť a nachádzať chyby v tom, čo vytvorili. Ďalej, pracovali na tom, ako majú veci správne fungovať. Sú nastavení na vykonávanie činnosti, ktorej výstupom sú pozitívne výsledky, pričom pri testovaní je potrebné hľadať zlyhania a negatíva. Ešte spomeniem to, že vývojári pracujúci v tíme majú prácu podelenú a každý zvyčajne pracuje na istej časti. Aby však mohol tester produkt kvalitne otestovať, musí vidieť produkt z oveľa širšej perspektívy. Ako tester som sa raz pozeral na to, ako sa vývojár snaží nasimulovať chybu, ktorú som predtým našiel a nahlásil mu. Bez podrobne popísaného postupu by mal problém cez používateľské prostredie vykonať celý proces a k danej chybe sa dostať. To, že pracoval len na istej časti funkcionality spôsobilo, že ju nepoznal ako celok a nevedel k nej pristúpiť z perspektívy používateľa.

Prečo firmy testujú manuálne?

Viem, že veľké softvérové firmy majú zvyčajne celé oddelenia, ktoré sa venujú tomu, ako vytvoriť čo najkvalitnejší produkt a v rámci toho majú zadané rôzne metódy na testovanie vytváraného produktu. Moje skúsenosti sú však z oblasti menších firiem.

Prvá je tá, že firmy zamestnávajú manuálnych testerov kvôli tomu, že iným spôsobom svoj vyvíjaný produkt netestujú. Nepraktizujú napríklad testami riadený vývoj, pri ktorom si vývojári najprv napíšu test a až potom programujú, aby keď dokončia nejakú ucelenú časť, si ju mohli po sebe otestovať. Dôvodom môže byť to, že chcú eliminovať čas vývojára, počas ktorého by si musel test vytvoriť a spúšťať ho. Spoliehajú sa na to, že v čase kedy vývojár programuje funkcionality produktu, môže tester zároveň produkt testovať.

Taktiež môžu byť firmy voči zautomatizovanému testovaniu nedôverčivé. Programovanie automatických testov, ktoré sú kvalitné a pokrývajú všetky prípady, ktoré má test overiť, si vyžaduje určité skúsenosti. Nie každý vývojár v malej začínajúcej firme ich má, a preto sa radšej zvolí taktika manuálneho testovania nad zautomatizovaným.

4 Filip Sucháč

Ďalej sa môže stať, že sa firma, resp. jej projekt ocitne v krízovej situácii kvôli zanedbaniu testovania. Tesne pred termínom akceptačného testovania sa zistí, že produkt nespĺňa požiadavky zákazníka a je nutné vykonať veľa zmien v krátkom čase. Táto časová tieseň produkuje mnohé chyby a najrýchlejším riešením môže byť povolanie viacerých testerov, ktorí budú paralelne produkt manuálne testovať a hlásiť chyby vývojárom.

Dokáže dostatočné množstvo testerov odhaliť všetky chyby?

Každý správny softvérový inžinier by na túto otázku nemal odpovedať slovom áno. Známý Dijkstrov výrok[2] hovorí, že testovanie nemôže preukázať, že v programe chyby nie sú, ale iba že tam sú. Preto sa nedá otestovať, či boli všetky chyby odhalené a odstránené. Je však logické, že bezchybnosť produktu by mala byť cieľom snahy každej firmy. Preto stratégia zamestnať čo najviac testerov je tiež logická, hoci ktovie či efektívna. Každý človek je iný a teda môže mať viac iných pohľadov a prístupov k danému problému. Je rozdiel medzi testerom študentom, testerom na plný úväzok a testerom, ktorý okrem toho aj programuje alebo vytvára dokumentáciu k projektu. Preto napríklad chybu, ktorú by jeden prieskumný tester v živote neodhalil, môže iný naopak odhaliť na prvý pokus. Teda kvantita testerov tu predsa len môže niečo znamenať, pretože s ňou prichádza aj ich užitočná rozdielnosť.

Záver

V predošlých riadkoch som sa zameril na manuálne testovanie, porovnal niektoré spôsoby testovania, opísal človeka testera a podelil som sa o niektoré skúsenosti z testovania softvéru. Na otázku, či má zmysel testovať softvér manuálne odpovedám určite áno. Myslím si, že dosiahnutie pozitív manuálneho testovania nahradením človeka strojom je ešte hudba ďalekej budúcnosti. Tento druh testovania však nejako špeciálne nevyzdvihujem. Vhodná cesta k dobrému testovaniu softvéru je podľa mňa v kombinácii prístupov spomínaných v tejto eseji. Určite som za testami riadený vývoj, za automatické testy a prieskumné testovanie. Ich miera použitia by však mala závisieť od povahy projektu a samotného vyvíjaného softvérového produktu. Správne odhadnúť túto mieru by mala byť naša motivácia neustáleho štúdia a skúmania tejto oblasti, zvanej testovanie.

Použitá literatúra

1. Armour, P. G.: *The Unconscious Art of Software Testing*. In: Communications of the ACM - Interaction design and children. Vol. 48, Issue 1, January 2005, 15-18.
2. Bieliková, M.: *Softvérové inžinierstvo – Princípy a manažment*. Vydavateľstvo STU, ISBN 80-227-1322-8, 2000.
3. Hamai, E.: *Software Testing: Is it necessary to have a good knowledge of programming to get into a testing career?* (2012)
<http://www.quora.com/Software-Testing/Is-it-necessary-to-have-a-good-knowledge-of-programming-to-get-into-a-Testing-career>

4. Martin, R. C.: *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008. 464 s. ISBN 0-13-235088-2.
5. Montvelisky, J.: *Why can't developers be good testers?* (2010)
<http://qablog.practitest.com/2010/05/why-cant-developers-be-good-testers/>

Annotation

Does it make sense to test software manually?

Every software company wants to develop a quality product. Specific part of processes that are dealing with achieving software quality is testing. There are lots of approaches to testing. In many companies exists a position called tester that is occupied by people who are testing product under development manually and reporting detected bugs. In this essay I am dealing with pros and cons of manual testing, describing what characteristics should good manual tester have and what approaches to testing are good to be chosen.