TESTOVANIE SOFTVÉROVÉHO PRODUKTU A JEHO VLASTNOSTÍ

„Niečo chcieť znamená už polovicu mať."

Tomáš Tomašovič

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
35153[zavináč]is[.]stuba[.]sk

Abstrakt. *V eseji je popísané všeobecné testovanie softvérového produktu, testovanie produktu ohľadom bezpečnosti, jeho funkcionality, správania sa v určitých situáciách. Softvérový produkt musí byť dôkladne zabezpečený proti nepovolanému prístupu do systému, úniku dát, či výpadku. Taktiež musí byť imúnny voči vstupu nekorektných a nekompletných dát od používateľa. Ďalej sa esej zameriava na rôzne techniky testovania softvérového produktu, charakterizuje viacero konkrétnych techník. Dôležitá je tiež kontrola kvality produktu, nedostatočne kvalitný produkt v budúcnosti vyžaduje ďalšie, mnohokrát zbytočné investície. V súčasnosti nie je kvalita a bezpečnosť softvérových produktov nie je na dostatočnej úrovni.*

Kľúčové slová: *softvér, softvérový produkt, bezpečnosť softvérového produktu, testovanie softvérového produktu, kvalita softvérového produktu.*

Úvod

V nasledujúcej eseji priblížim pojem testovania, jeho dôležitosť pri vývoji softvéru a jeho vplyv na kvalitu výsledného produktu.

Je samozrejmé, že testovanie softvérového produktu je nevyhnutné na zabezpečenie jeho kvality a správnej funkcionality. Je to súbor procesov na kontrolu kvality produktu [4]. Testovanie sa môže aplikovať na celý produkt, alebo len na jeho súčasti. Existuje viacero postupov a spôsobov testovania na kontrolu požadovaných vlastností a správania

sa výsledného produktu. Vlastnosti produktu, ako sú jeho bezpečnosť, funkčnosť, spoľahlivosť, výkonnosť, použiteľnosť, podporovateľnosť, sú dôležitými súčasťami každého systému. Zanedbanie čo i len jednej vlastnosti by skôr či neskôr viedlo k nespokojnosti zákazníka a tým pádom k finančným stratám, či už priamym, alebo nepriamym. Samotné testovanie tieto riziká nevylúči, ale minimalizuje. Napriek tomu sa tomuto procesu ani v súčasnosti neprikladá veľká dôležitosť z viacerých dôvodov. Jedným z nich je ušetrenie finančných nákladov, nedostatok času, či len lenivosť.

Vlastnosti softvérového produktu sú definované normou ISO/IEC 9126. V nasledujúcej tabuľke vidíme tieto vlastnosti popísané s čiastkovými atribútmi [5].

Tab. 1. Vlastnosti a čiastkové atribúty kvality definované normou ISO/IEC 9126

Vlastnosť	**Čiastkové atribúty**
Funkcionalita	Vhodnosť, presnosť, interoperabilita, bezpečnosť.
Spoľahlivosť	Vyzretosť, tolerancia k chybám, obnoviteľnosť.
Použiteľnosť	Porozumiteľnosť, naučiteľnosť, funkčnosť, atraktívnosť.
Efektivita	Časová efektivita, využitie zdrojov, naplnenie efektivity.
Udržovateľnosť	Analyzovateľnosť, meniteľnosť, stabilita, testovateľnosť.
Prenositeľnosť	Prispôsobivosť, inštalovateľnosť, nahraditeľnosť, spolužitie.

Testovanie softvéru

Čo je vlastne testovanie ? Existuje viacero definícií, jedna z nich napríklad hovorí:

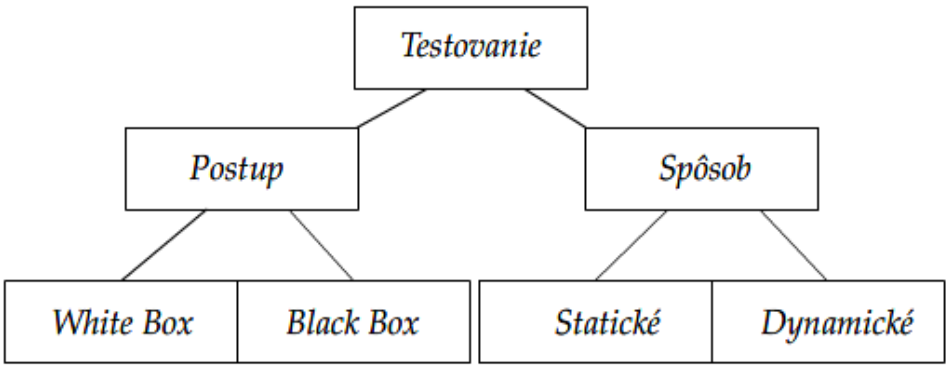
„Testovanie je plánovaný a systematický proces, ktorý zahrňuje fázy úzko spojené s fázami vývoja IT riešení. Od tejto úzkej spojitosti závisí, aký druh testu bude použitý, aký mechanizmus, techniky a požadované zdroje by mali byť použité počas cyklu testovania.

Testovanie nemôže nikdy kompletne zaručiť korektné správanie softvéru. Namiesto toho pristupuje k softvéru kriticky a hľadá v ňom chyby (defekty) alebo porovná stav a správanie softvérového produktu oproti špecifikácii [1].“

Myslím, že táto definícia nám dostatočne popisuje proces testovania softvérového produktu. Avšak existuje ešte ďalšia problematika súvisiaca s testovaním.

V úvode bol spomenutý napríklad postup a spôsob testovania. Stručne opíšem obe metódy, ako sú znázornené na obrázku 1. Samotný postup pozostáva z dvoch metód a to z White Box a Black Box testovania. Pri Black Box testovaní používateľ nemá prístup k zdrojovým kódom programu, ale vie, čo má testovaný program robiť. Naopak, pri White Box testovaní má prístup k zdrojovým kódom programu.

Existujú dva spôsoby testovania, statické a dynamické testovanie. Pri statickom testovaní sa testuje celkový produkt, kontroluje sa zdrojový kód. Pri dynamickom testovaní sa produkt kontroluje a testuje popri jeho vývoji, spadá sem aj napríklad záťažové testovanie.



Obr. 1. Rozdelenie testovania podľa spôsobu a postupu.

Záťažové testy dobre popisuje nasledujúca definícia:

„Záťažové testy (load tests) patria medzi vysokoúrovňové testy, ktoré simulujú prácu veľkého počtu užívateľov na testovanej aplikácii alebo generujú veľký počet operácií v informačnom systéme.

Súčasťou záťažového testovania je meranie výkonu a odozvy aplikácie v závislosti od počtu užívateľov súčasne pracujúcich s danou aplikáciou.

Na základe analýz z jednotlivých meraní sa identifikujú problémy vzniknuté záťažou, následne sa môžu vykonať opravy, upgrady alebo optimalizácie aplikácie“ [2].

Dodám, že záťažové testovanie je najčastejšie vykonávané automaticky. Manuálne testovanie softvéru majú najčastejšie na starosti testeri. Simulujú správanie používateľa a vedia chybu podrobne opísať, prípadne určiť jej lokalizáciu v programe. K testerom sa vrátim ešte neskôr v eseji.

Testovaním sa preverujú nasledovné vlastnosti softvérového produktu [6]:

- funkčnosť (úžitkovosť, utility),
- spoľahlivosť (reliability),
- robustnosť (robustness),
- výkonnosť (performance),
- správnosť (korektnosť, correctness)

Každá z vlastností je dôležitá pre korektné fungovanie produktu. Celý proces testovania je etapovitý a zahŕňa [6]:

- testy funkcií a modulov, ktoré sú vykonávané prevažne vývojármi priamo po fáze implementácie,
- integračné testy, t. j. vlastné testovanie viacerých modulov súčasne,
- nezávislé testy – vykonávané nezávislými externými skupinami,
- inštalačné testy – zahŕňajú všeobecný výkon systému, ktorý je prvýkrát nainštalovaný na konkrétnom operačnom systéme,
- alfa a beta testovanie – t. j. testovanie systému v reálnom prostredí (nie u vývojárskej firmy). Pri alfa testovaní sa systém testuje bez reálnych dát. Beta testovanie používa reálne dáta so sledovaním výsledkov,

- preberacie testovanie – t. j. posledný míľnik pri testovaní produktu. V prípade úspešného zvládnutia nastáva oficiálne prevzatie produktu zákazníkom.

V týchto bodoch sme sa dozvedeli etapy testovania softvéru, ale je ešte dôležité spomenúť niekoľko zásad pri testovaní.

- zamerať sa na verifikáciu a validáciu všetkých fáz vývoja, t. j. koncepciu, požiadavky, riešenie, implementáciu, testy, inštaláciu, prevádzku a údržbu,
- štandardizované postupy (umožňujú redukovať námahu pri písaní a protokolovaní),
- rôzna hĺbka skúšok v závislosti od objektu,
- opakovateľnosť,
- vylúčenie subjektívnych vplyvov,
- detailná skúšobná dokumentácia (ktorá umožňuje skúšku opakovať),
- časové usporiadanie postupu (predbežná kontrola, skúšanie, skúšanie zmenených častí),
- logické usporiadanie postupu (zoskupenie podobných funkcií a ich kontrola, kontrola jednotlivých samostatných častí systému, interakcia medzi subsystémami, celková kontrola),
- hierarchické usporiadanie postupu (týka sa dokumentácie).

Dodržanie týchto zásad sa v praxi veľmi oplatí, keďže v konečnom dôsledku sa ušetrí čas a prostriedky. Čo sa týka času testovania, tak časová náročnosť skúšok bola stanovená empiricky. Orientačne platia hodnoty z tabuľky 2, ale osobná skúsenosť autorov s vývojom komerčných informačných systémov hovorí, že predpokladané časy testovania sú vždy poddimenzované [6].

Tab. 2. Odhady časovej náročnosti testov

Testovaný objekt	Časová náročnosť testov	Stredná hodnota
Vstupy a výstupy (vstupné a výstupné polia)	0,5 – 2,0 hod	1,0 hod
Elementárne funkcie (základné operácie a činnosti bez potreby volania iných funkcií)	0,5 – 5,0 hod	3,0 hod
Vyššie funkcie (s potrebou výpočtov, volajúce iné funkcie, s potrebou prístupu do dátových súborov a pod.)	5,0 – 100,0 hod	10,0 hod
Komplexné funkcie	Suma časov potrebných na otestovanie funkcií, ktoré v nich vystupujú	

Odhad podielu časov jednotlivých činností pri procese testovania je približne nasledovný [3]:

- Odhad náročnosti skúšok 1 %
- Kontrola používateľskej dokumentácie 22 %
- Riadenie a koordinácia testovania 15 %
- Vytvorenie a dopĺňanie skúšobných plánov 18 %
- Testovanie konzistencie programu, dokumentácie a opisu produktu 15 %
- Testovanie programu 28 %
- Vytvorenie skúšobnej správy 1 %

Ako vidieť, podiel samotného testovania programu nie je vôbec zanedbateľný.

Bezpečnosť

Nemenej dôležité testovanie je testovanie bezpečnosti aplikácie. Toto testovanie pomáha odhaľovať chyby, ako je nepovolený prístup k určitým dátam, informáciám, alebo len znepríjemnenie práce ostatným používateľom. Táto problematika sa týka hlavne webových aplikácií, pri desktopových aplikáciách, ku ktorým má fyzický prístup obvykle len jeden používateľ, je riziko menšie. Vždy tu však to riziko je.

Používatelia snažiaci sa o prístup do systému sa často označujú ako „hackeri". Títo používatelia majú častokrát nadpriemerné znalosti o použitej technológii, vedia, ako sa správa v rôznych situáciách, a dokážu správanie aplikácie čiastočne alebo úplne ovládať. Ich motiváciou je nielen získanie dát a prístupu, ale často aj finančný prospech zo získaných informácií. Týmto konaním spôsobia firme škodu, alebo poškodia dobré meno firmy. Z tohto vyplýva dôležitosť testovania softvérového produktu na bezpečnosť, najčastejšie penetračnými testami.

Penetračné testovanie je testovanie aplikácie na vniknutie k dátam. Častokrát ho vykonávajú práve hore spomenutí hackeri. Takéto testovanie v sebe zahrňuje viacero techník prieniku. Charakterizujem dve najznámejšie chyby, ktoré umožňuje bezpečnostné testovanie odhaliť:

- SQL injection : pomocou tejto metódy môže útočník ovládať správanie SQL databázy a donútiť ju napríklad zobraziť dáta, ku ktorým používateľ normálne nemá prístup (heslá, registračné údaje a pod.). Chyba spočíva v nedostatočnom ošetrení vstupov od používateľa.
- XSS : Táto metóda útoku môže byť využitá rozličnými spôsobmi. Najčastejšie sa pomocou nej kradnú informácie, ktoré sú dôležité k prihláseniu používateľa. Nejde o kombináciu mena a hesla, ale o jednorazový, časovo ohraničený kľúč prihlásenia. Podľa druhu chyby umožňujúcej XSS je možné na stránky vkladať nežiadúce reklamy, spamy, alebo vytvárať dôveryhodný phishing (kradnutie prihlasovacích údajov pomocou falošnej stránky). Ochrana proti tomuto útoku je vcelku jednoduchá, stačí automaticky ošetriť každý vstup od používateľa.

Testovanie bezpečnosti patrí podľa mňa medzi najdôležitejšie druhy testovania softvérového produktu, pretože jeho zanedbanie by neviedlo iba k finančným stratám, ale i k strate dobrej povesti firmy, ktorá produkt vyvinula.

Funkcionalita

Testovanie funkcionality spočíva v kontrolovaní správania programu, či robí to, čo má robiť. Dá sa pod tým predstaviť napríklad postupnosť nejakých úkonov, kde záleží na poradí stlačenia tlačidiel. Mnohokrát sa stane, že používateľ postláča rôzne ovládacie prvky programu, čo program zmätie a vedie k jeho pádu, alebo zmrznutiu. Preto treba pri vývoji programu dbať aj na logickú náväznosť ovládania a funkcionality programu, aby nedochádzalo k spomínaným problémom.

Toto testovanie majú väčšinou na starosti testeri, ktorí poznajú problematiku, ktorú produkt rieši, ale nepotrebujú poznať, ako program funguje. Títo testeri skúšajú používať program na „nečisto" a hlásiť nezrovnalosti v logike produktu, nie v jeho kóde. Takéto testovanie nie je o nič menej dôležité, ako ostatné druhy testovania softvérového produktu, pretože napriek správne napísanému programu vyžaduje zmena časti logiky niekedy oveľa viac času, ako opravenie chyby v kóde.

Používateľ

Netreba zabúdať ani na potreby používateľa. Vstupnou bránou do aplikácie je pre neho používateľské rozhranie – GUI. Práve používateľské rozhranie môže celý softvérový produkt „potopiť" napriek tomu, že samotný produkt funguje bezchybne. Používateľ nemusí vždy byť technicky zdatný, aby vedel, čo robia niektoré exotické ovládacie prvky v programe, potrebuje sa rýchlo naučiť s aplikáciou pracovať, nie ju študovať. Preto je vhodné dbať aj na jednoduchosť používateľského rozhrania a pravidelne testovať, aké náročne je v ňom spraviť konkrétnu úlohu. Dobrým príkladom je kontrola údajov od používateľa, keď do vstupného prvku pre čísla vloží písmeno, alebo niečo podobné. Netestovaná aplikácie by v tom horšom prípade spadla a zobrala so sebou všetky neuložené údaje. Ideálne by mala upozorniť používateľa na chybu počas písania údaju do vstupného formulára. Toto testovanie je jedným z mnoho druhov testovaní používaných v praxi.

Záver

V eseji som sa zameral na problematiku testovania, zo začiatku som teoreticky opísal, čo to testovanie je, jeho rozdelenie a rôzne druhy testovania. Porovnal som definície z rôznych zdrojov, taktiež som popísal dopad testovania na kvalitu softvéru.

V ďalšej časti eseje som sa venoval bezpečnosti softvérového produktu, ako sa produkt testuje ohľadom bezpečnosti pomocou penetračných testov. Taktiež som charakterizoval dve najbežnejšie metódy prieniku a ich ošetrenie.

Pozornosť som venoval aj používateľovi, ktorý môže nevedome vyvolať chyby v programe a následne stratiť dôveru v daný produkt. Nemenej dôležité je aj testovanie používateľského rozhrania - GUI, ktorého otestovanie som tiež spomenul.

Použitá literatúra

1. Hanko M.: *Testovanie softvéru*, Dostupné na internete: http://butteland.blogspot.com/2008/08/testovanie-softvru.html, [cit: 2009-October]
2. Hanko M.: *Záťažové tesovanie*, Dostupné na internete: http://butteland.blogspot.com/2008/09/aov-testovanie.html, [cit: 2009-October]
3. Lindermeier, R., Siebert, F.: *Soft wareprűfung und Qualitätssicherung*. Springer-Verlag, Műnchen 1997.
4. RuleWorks: *The test managment guide*, Dostupné na internete: http://www.ruleworks.co.uk/testguide/, [cit: 2009-October]
5. Scalet et al., 2000: *ISO/IEC 9126 and 14598 integration aspects: A Brazilian viewpoint*. The Second World Congress on Software Quality, Yokohama, Japan, 2000.
6. Tanuška,P., Schreiber, P.: *Poznámky k problematike testovania informačných systémov*, Dostupné na internete: http://www.atpjournal.sk/casopisy/atp_99/atp10/tanuska.htm, [cit: 2009-October]

Annotation

Testing software product and its properties

The following essay describes the general software project testing and the safety of the product, its functionality, behavior in certain situations. The project must be properly secured against unauthorized system access, data leakage or loss. It must be immune to the entry of unfair and incomplete data from the user. Furthermore, the essay focuses on various techniques for testing software product, characterized by a number of specific techniques. Also important is the project quality control, product with a lack of quality often needs unnecessary investment in the future. Nevertheless, the quality and safety of software products is not sufficient even today. The essay is guided by personal observations and opinions.