

# MOŽNOSTI ZNÍŽENIA NÁKLADOV PRI TESTOVANÍ WEBOVÝCH APLIKÁCIÁCH

*„Testovanie pomocou koncových používateľov je už praktickým štandardom, pomocou tej metódy je možné odhaľovať veľké množstvo chýb.“*

*András Lendvay*

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

**Abstrakt.** V dnešnej dobe webové aplikácie tvoria významnú časť softvérových produktov. Viac a viac podnikov potrebuje webový systém na zabezpečenie obchodných služieb. Bez použitia webových systémov podniky nemajú šancu na úspešné fungovanie na trhu. Webové systémy sa neustále vyvíjajú, sú čím ďalej tým zložitejšie, preto zabezpečenie kvality môže byť väčším problémom, ako sa to na prvý pohľad zdá. Testovanie môže zabrať viac času, a tým pádom aj náklady môžu rásť do nevidaných rozmerov. V tejto eseji sa zaoberám možnosťami šetrenia nákladov pri testovaní webových softvérových systémov, s možnosťami rôznych metód testovania pomocou koncových používateľov a s možnosťami automatickej detekcie chýb. Zamieriam sa aj na možnosti, ako motivovať koncových používateľov k testovaniu produktu v rámci beta testov.

**Kľúčové slová:** testovanie, beta testy, automatické testovanie, kvalita, webové softvérové systémy

## Úvod

Webové systémy sú už základom ponuky firiem, preto ich kvalita sa stáva viac a viac dôležitou. Používatelia majú možnosť porovnať si viacero webových stránok s podobným obsahom a jednoducho si môžu vyberať medzi nimi. Aj najmenšie chyby alebo nefunkčnosť môžu znamenať stratu zákazníka. Konkurencia je veľmi silná, práve preto je dôležitá kvalita softvérového produktu pre zákazníka. Firma s nekvalitným softvérom bude vytlačená z trhu pomocou kvalitného softvéru iného podniku.

Najdôležitejšou vlastnosťou kvalitného webového softvéru je použiteľnosť. Používatelia webových systémov chcú pracovať v systéme bez potreby školenia. Z uvedeného sa vyplýva, že je dôležité, aby používateľ mal všetko jasné na prvý pohľad, aby našiel všetko ľahko, čo potrebuje. Dobře vyvinuté používateľské rozhranie je základnou časťou kvalitného webového produktu. Problém je v tom, že testovanie použiteľnosti je možné len pomocou koncových používateľov.

Testovanie pomocou koncových používateľov je už praktickým štandardom, pomocou tejto metódy je možné odhaľovať veľké množstvo chýb. Jeho jedinou nevýhodou je, že potrebujeme k testovaniu už takmer hotový softvérový systém, produkt, ktorý je už spustiteľný a poskytuje najdôležitejšiu funkcionálnosť.

Webové aplikácie majú jednu veľkú výhodu proti desktopovým aplikáciám. Webové systémy sú umiestnené na jednom mieste, preto nie je potrebné zmeniť aplikáciu každej inštalácie na každom počítači. Webové aplikácie sú preto oveľa flexibilnejšie a táto vlastnosť nám umožňuje vykonať zmeny s minimálnym úsilím.

## Možnosti testovania

Na testovanie existuje veľa klasických prístupov, najtypickejší je testovanie v sledovanom prostredí. Táto metodológia nie je úplne vhodná na testovanie webových softvérových systémov. Je limitovaná a potrebuje platených testerov. Analýza je časovo náročná, je potrebné kontrolovať všetky akcie používateľov. Preto klasický prístup je limitovaný, časovo náročný a žiada veľké množstvo nákladov.

Z tohto dôvodu boli vytvorené nové metódy, ktoré sa dajú použiť aj pri testovaní webových systémov. Také metódy sú heuristické hodnotenie [1] a „A/B testing“ [2], ktoré môžu znížiť náklady pre celý proces vyhodnotenia. Tie metódy pomôžu identifikovať problémy spojené s použiteľnosťou. V praxi sa bežne používajú a môžu šetriť náklady, ako aj čas testovania.

Treba pripomenúť, že efektivita testov je jednoznačná pri inkrementálnom vývoji softvéru, lebo odhalenie chýb je zamerané na testovanie skoro hotového systému. Ak použijeme uvedenú metódu testovania s vodopádovým modelom, väčšinu chýb budeme schopní identifikovať len pri konečnej fáze vývoja. T tohto dôvodu pravdepodobne nebudeme schopní dodržať termín odovzdania projektu. Pri inkrementálnom vývoji môžeme kontrolovať chyby aj v systémoch, kde sú pridané ešte len základné funkcionality.

## Testovanie pomocou používateľa

Testovanie pomocou používateľov môžeme rozdeliť na dve základné skupiny :

1. testovanie sa uskutočňuje v sledovanom prostredí a je dostupná neustála komunikácia s používateľom, túto metódu nazývame alfa testovanie
2. druhou skupinou je beta testovanie, kde používateľ testuje v svojom prostredí a systematicky posiela svoje pripomienky

Obidva testy majú svoje výhody a nevýhody, pri alfa testovaní existujú zvyčajné postupy, podľa ktorých sa má používateľ riadiť. Tie testy sú zvyčajne drahšie, preto máme k dispozícii len limitovaný počet používateľov a testov. Pri beta testovaní je k dispozícii viac testov a tým pádom aj viac dát, ale za cenu. Používateľov musíme buď platiť, alebo motivovať ich k testovaniu nášho produktu. Nevýhoda beta testovania je limitovaná komunikácia s používateľom.

## Máme platiť beta testerov?

Odpoveď na túto otázku nie je tak jednoduchá, môže závisieť od oblastí a typu aplikácie, ktorú testujeme. Je možnosť mať platených testerov, ktorí majú aj vedomosti v danej oblasti. V prípade, ak chceme opraviť veľké množstvo chýb, ten prístup nie je efektívny, lebo náklady sa budú značne zvyšovať zvýšeným počtu testerov.

Otázka, či máme k dispozícii dostatok bezplatných používateľov na testovanie, závisí najmä od vzťahu medzi používateľom a softvérovým systémom. Systém môže byť prostriedkom určitej komunity a softvér používajú každodenne v rámci svojich prác. V takom prípade máme ľahkú úlohu, lebo používatelia majú záujem o vylepšenie produktu. Iný typ produktu je, ak koncoví používatelia nemajú záujem o kvalitu softvérového systému. Jasným príkladom môže byť napríklad webový obchod, ktorý používatelia nepoužívajú každodenne pre svoju prácu, len raz na objednanie tovaru. Ak používateľ nejaví záujem o kvalitu produktu, softvér sa nedá testovať pomocou bezplatných používateľov. Dôležitý pre typ systému, či hovoríme o unikátnom produkte, lebo v opačnom prípade pravdepodobne nebude používateľov testovanie zaujímať. Ale ak sa jedná o softvérový systém, ktorý je revolučne nový a úplne jedinečný, používatelia budú chcieť si vyskúšať aj náš priebežný produkt. V praxi je ťažké ponúkať používateľovi niečo ozaj zaujímavé, aby použili náš priebežný produkt. Vývojári často majú problém aj s presvedčením používateľov k používaniu ich konečného produktu. Záujem o testovanie môžeme zvyšovať s rôznymi spôsobmi. Jednou možnosťou je, ak sa produkt bude predávať za úhradu, potom môžeme ponúkať vyskúšanie produktu zadarmo počas beta testov. Treba si uvedomiť, že v niektorých prípadoch je priam nemožné používať bezplatných testerov, hlavne ak systém nie je unikátny, ponúka len základnú funkcionálnosť, jeho použitie nie je zábavné pre používateľa. V tom prípade sa budeme musieť spoliehať na služby platených testerov.

Opačným problémom je, ak máme v dispozícii veľké množstvo používateľov, ktorí chcú testovať náš produkt bezplatne. Keby sme pripustili každého používateľa k testovaniu nášho produktu, dostali by sme množstvo dát, ktoré už pravdepodobne nedokázali spracovať. Obstaranie dostatočných zdrojov by bolo tiež len ťažko splniteľné.

Možnosť vyskúšania produktu ponúkané pre každého má aj iné nevýhody. Mnohí používatelia si neuvedomujú, že verzia produktu nie je ešte finálna a tým pádom označia ju ako nekvalitnú. Ak sa neskôr stretnú s hotovým produktom, spomenú si len na problémy a nedostatky a tým pádom stratí popularitu nielen produkt, ale aj firma, ktorá vytvorila systém. Z tohto dôvodu rozdiel medzi beta verziou a konečným produktom treba vždy definovať. Najlepší prístup teda je vybrať si používateľov podľa daných kritérií. Najdôležitejšia podmienka je, aby naši tester skúsili rovnaké funkcionality, ako naši koncoví používatelia. Nakoľko používatelia testujú náš produkt bezplatne, môžeme predpokladať, že každý z nich má záujem použiť aj náš konečný produkt, preto naši tester majú rovnaké vlastnosti, ako koncoví používatelia. Preto by sme si mali vybrať používateľov, ktorí nebudú len pasívne používať zatiaľ neukončený produkt, ale pokúsia sa komunikovať s vývojármi.

### **Komunikácia s testujúcimi používateľmi**

Komunikácia s používateľmi, ktorí testujú náš systém, je nevyhnutná aj preto, aby sme získali informácie o problémoch v našom softvérovom systéme. Na komunikáciu s vývojármi môžeme definovať tri praktické možnosti:

1. vyplnenie formulára
2. písanie na fóra
3. vytvorenie reportov

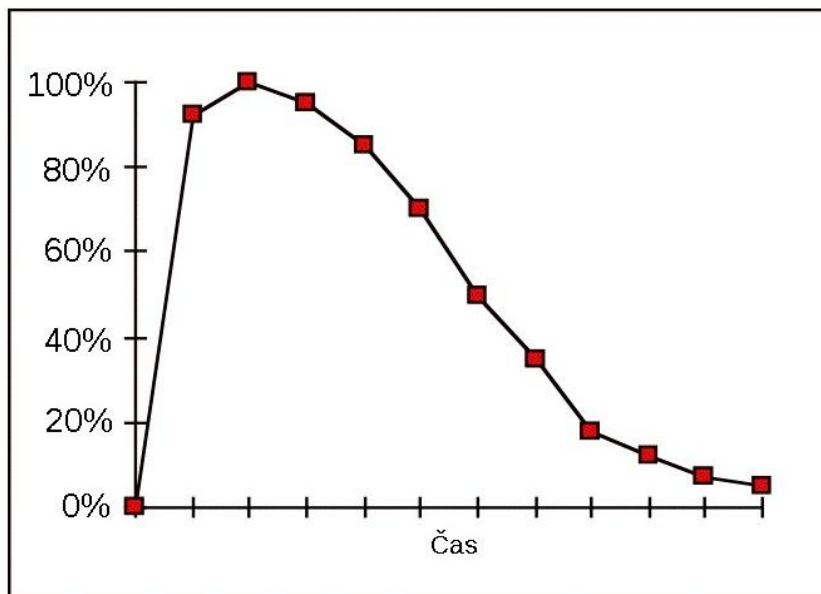
Každá z tých metód má svoje výhody, ale aj nevýhody. Formuláre sú rýchle, tým pádom je väčšia šanca, že aj bežný tester ich vyplní. Tie formuláre používateľ vyplní v prípade, ak našiel nejakú chybu, kde môže opísať daný problém alebo po ukončení funkcionality, kde používateľ môže napísať názor a hodnotiť funkcionality. Nevýhoda formuláru je, že treba ju implementovať do systému. Niektorí používatelia radšej píšu chyby na fóra, kde pravdepodobne rýchlo dostanú spätnú väzbu od iných používateľov, alebo od vývojárov produktu. Nevýhodou tejto metódy je veľké množstvo vytvorených tém, ktoré je potom ťažké kategorizovať a vyhodnotiť podľa vážnosti problémov. Tretou možnosťou je písanie reportov, kde používateľ môže dobre opísať danú chybu. Nevýhodou je potreba vedomostí používateľov v oblasti, neskúsení používatelia môžu poslať veľa falošných informácií. Ľudskou vlastnosťou používateľov je, že sú leniví písať viacstranové reporty, preto táto metóda je vhodnejšia pri platených testeroch. Na základe vlastností formulára a fóra sú najdosvedčenejšie metódy pri testovaní pomocou používateľov.

Podľa informácií doručené od používateľov môžeme identifikovať rôzne chyby. Tie chyby treba hodnotiť a vytvoriť poradie dôležitosti, na základe čoho budú zoradené na následnú opravu. V prvom kroku by mali opravovať chyby, ktoré majú vplyv na základnú funkcionality. Priorita opravovania chýb by mala závisieť aj od počtu testujúcich používateľov, ktorým sa daná chyba vyskytla.

### **Motivácia testujúcich používateľov**

Záujem testujúcich používateľov klesá časom (vid'. obr. 1), preto je nutné vytvoriť rôzne metódy na ich motiváciu. Treba ich motivovať, aby používatelia použili stránku každý

deň, tým je oveľa väčšia šanca nájdania chýb. Treba ich motivovať, aby dali aj spätnú väzbu o funkcionalite systému a nie len ju používali.



**Obr. 1.** Motivácia bezplatných používateľov podľa času.

Na motiváciu je osvedčená metóda vrátiť informáciu o stave stránky, aby používateľ cítil, že je časťou komunity a nie len jeden z viacerých testerov. Keď používateľ ohlási chybu, je dôležité mu dať spätnú väzbu (aj keď je to automaticky vygenerovaná správa), aby videl, že jeho práca má vplyv na výsledný produkt. Je dôležité, aby vývojári písali na fóra, ak daný problém je vyriešený a poďakovali sa používateľom za nahlásenia chýb. V praxi riešia nemotivovaných používateľov pomocou spustenia viacerých vln na prijatie nových testerov, tým môžu znova obnoviť radu používateľov [3].

### Automatické detekovanie chýb

Popri testovaní softvérového systému pomocou používateľov môžeme použiť aj metódu automatického testovania. Testovanie pomocou automatického detekovania chýb má svoje výhody a nevýhody oproti testovaniu pomocou používateľov. V praxi nie sú tak presné ako testovanie pomocou používateľov, predsa vykonáme veľké množstvo testov za malé náklady, tým môžeme odhaliť veľa chýb. Nevýhodou tohto testovania, že nie je kompletná, a pre nájdanie niektorých chýb potrebujeme ľudí (napríklad na hodnotenie používateľnosti, hlavne na ohodnotenie používateľského rozhrania).

Základnou myšlienkou automatického testovania je kontrola kódov pomocou rôznych algoritmov. Existujú už technológie na automatické testovanie, aké sú EXE [5] a Dark [6], ale tie nie sú použiteľné pri webových systémoch. Pri webových systémoch môžeme ohodnotiť statické webové stránky (technológie CSE HTML Validator a W3C HTML validator), ale pri vyhodnocovaní dynamických stránok sa vyskytnú rôzne problémy. Pri dynamických stránkach (hlavne pri jazyku PHP), dynamické stránky

vygenerujú statické HTML stránky. Tie stránky môžeme kontrolovať, či neodkazujú na obsah, ktorý nie je dostupný, alebo môžeme kontrolovať ich syntax. Syntakticky nekvalitné stránky môžu vytvárať problémy pri niektorých webových prehliadačoch a môžu vytvoriť aj bezpečnostné problémy. Statické validátory pri hodnotení dynamických stránok sa už používajú v praxi, ale majú určité nedostatky.

Keďže statické testovanie má nedostatky, v posledných rokoch sa vyvíjajú nové technológie na ohodnotenie dynamických stránok. Podstatou týchto metód je spracovanie údajov získaných zo sledovania činností používateľov na webovej systéme. Dáta vytvorené pri logovaní môžeme použiť na vytvorenie testovacích úloh a pomocou predpovede (oracle) vyhodnotíme, či aktuálne výstupy sú zhodné s očakávanými výstupmi [7]. Ľudský tester by mohol urobiť aj aplikáciu testov, aj predpovede, ale je to veľmi únavné a s veľkou pravdepodobnosťou by sa vyskytli chyby. Pri automatických testovaniach treba riešiť viac problémov. Napríklad pri vytvorení úloh treba kontrolovať, či použité funkcionality nepotrebovali interakciu viacerých používateľov. Iným problémom môže byť, že stav servera sa neustále mení a predchádzajúce testovacie vstupy môžu vytvárať iné výstupy. Časť problémov pri daných vstupov nie je možné odhaliť. Napriek tomu časť vstupných údajov môžeme vyhodnotiť úspešne, môžeme odhaliť veľké množstvo chýb. Je nutné pripomenúť, že na automatické testovanie potrebujeme odborníkov, ktorí pracujú v danej oblasti. Funkcionality automatického testovania ešte nie sú úspešne používané v praxi, nakoľko sú zatiaľ vo výskumu.

Keď si chceme zhrnúť výhody a nevýhody automatického testovania, môžeme jednoznačne tvrdiť, že tento typ testovania je výhodný, lebo je oveľa lacnejší. Môže odhaliť veľké množstvo chýb, ale je limitovaný, lebo nemôže odhaliť nedostatky z ohľadu používateľnosti. Ohraničenosť nie je až takým veľkým problémom, lebo nedostatky môžeme odhaliť s využitím testovania pomocou používateľov. Pre ohodnotenie statických webových stránok máme možnosť aj pri výbere prostriedkov a výsledky môžu byť viac ako dostatočné. Pri dynamických stránkach metódy sú ešte vo fázy vývoja, najmä testované pri laboratórnych podmienkach, ale za pár rokov už môžu vyskytnúť silné prostriedky na testovanie dynamického obsahu webových systémov.

## **Záver**

V eseji som sa zamerlal najmä na problematiku testovania webových systémov. Opísal som základné metódy zabezpečenia kvality a testovania, pokúsil som sa pravidlá aplikovať takým spôsobom, aby náklady boli nižšie proti klasickým testovaním.

V eseji som analyzoval možnosť testovania pomocou koncového používateľa, vymenoval som možnosti ako získať bezplatných používateľov, aby náklady počas testovania sa nezvyšovali. Písal som o niektorých problémoch, ktoré sa môžu vyskytnúť počas beta testov a vyhodnotil som možné riešenia. Skúsil som nájsť optimálne vlastnosti, podľa ktorých sa môže uskutočniť výber beta testerov. Definoval som základné možnosti komunikácie medzi vývojármi a testerami. Metódy som ohodnotil podľa použiteľnosti pri bezplatných testeroch. Motivácia bezplatných testerov klesá rýchlo, preto som vymenoval pár možností, ako testujúcich používateľov motivovať.

V druhej časti eseje som opísal možnosti zníženia nákladov pomocou detekovania chýb. Vyhodnotil som už existujúce metódy automatického testovania, opísal som statické

metódy použité v praxi a hodnotil som výhody a nevýhody automatického testovania oproti testovaniu pomocou koncových používateľov.

Na otázku „Ako môžeme znížiť náklady pri webových aplikáciách s dodržaním kvality?“ by som zhrnul, že pre dosiahnutie dostatočne funkcionálneho systému treba použiť testovanie pomocou vývojárov. Ak systém je blízko konečného stavu, potom môžeme použiť automatické testovanie na softvérovom systéme. Keď systém už má hotovú funkcionálnosť a hlavne chceme otestovať používateľnosť a odhaliť menšie chyby, môžeme vstúpiť do fázy alfa a beta testovania. Tajomstvom dobre a lacno otestovaného systému sú správne zvolené hranice medzi rôznymi testami.

## Použitá literatúra

1. Nielsen, J. and Molich, R. Heuristic evaluation of user interfaces. Proceedings of CHI 1990, ACM Press (1990), 249-256.
2. Au, I., Boardman, R., Jeffries, R., Larvie, P., Pavese, A., Riegelsberger, J., Rodden, K., and Stevens, M. User experience at google: focus on the user and all else will follow. Proceedings of CHI 2008, ACM Press (2008), 368-3686.
3. Joel Spolsky Top Twelve Tips for Running a Beta Test, March 02, 2004
4. Bertolino, A. 2007. Software Testing Research: Achievements, Challenges, Dreams. In 2007 Future of Software Engineering (May 23 - 25, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 85-103.
5. C. Cadar, V. Ganesh, P. M. Pawlowski, D. L. Dill, and D. R. Engler. EXE: automatically generating inputs of death. In CCS, 2006.
6. P. Godefroid, N. Klarlund, and K. Sen. DART: Directed automated random testing. In PLDI, 2005.
7. Sprenkle, S., Gibson, E., Sampath, S., and Pollock, L. 2005. Automated replay and failure detection for web applications. In Proceedings of the 20th IEEE/ACM international Conference on Automated Software Engineering (Long Beach, CA, USA, November 07 - 11, 2005). ASE '05. ACM, New York, NY, 253-262. DOI=<http://doi.acm.org/10.1145/1101908.1101947>

## Annotation

*Possibilities of cutting down expenses during the testing of web-based applications*

*Nowadays web-based applications are an important part of software systems. They are needed by more and more companies to ensure business services. Web-based systems are constantly in development and as they are more complex, the assurance of their quality can cause major problems. Testing can require a high amount of time, so the expenses can grow to unexpected measures. In this essay I am going to be concerned of possibilities of cutting down expenses during the testing of web-based applications, with the possibilities of*

**8** *András Lendvay*

*different methods of user-based tests and with the possibilities of automatic error detection. I am also going to study different ways of motivation users during user-based testing.*

*Keywords: software testing, beta tests, automatic testing, software quality, web-based software systems*