

TESTIRANJE PROGRAMOV - KAJ PA PODATKI?

Franc Žerdin

Kljub vsej naglici sodobnega poslovnega življenja je prav, da se od časa do časa ustavimo, se ozremo nazaj in ugotovimo, kaj smo in česa nismo storili na opazovanem področju.

Nekaj takega so storili uredniki revije *Communications of ACM* (april, 1997), ki so glavno temo (7 člankov) posvetili testiranju programov. Gostujoči urednik (M. Lieberman) uvodoma ugotavlja, da kljub vsemu napredku v zadnjih 30. letih - zmogljivejši in hitrejši računalniki, omrežja, grafični vmesniki in vse drugo - obstaja pri razvoju programske opreme nekaj neprijetnih, zoprnih dejstev, ki jih še vedno nismo odpravili. Večina teh zoprnih dejstev leti seveda na pomanjkljive tehnike testiranja in nezadostno uporabo avtomatskih orodij pri testiranju.

Tudi zagotavljanje zanesljivih in točnih podatkov je še vedno daleč od popolnosti. In vendar uporabniki niso preveč zainteresirani za razvoj programske opreme, dokler ta pravilno deluje, točnim in zanesljivim podatkom pa se nočejo in tudi ne morejo odpovedati. Pri obravnavanju velikih količin podatkov problemi ne le naraščajo, ampak postanejo kompleksnejši in težje obvladljivi. Pri tem obstaja še ena ovira, namreč, da so tovrstne izkušnje običajno zapisane le v internih poročilih posameznih družb.

Testiranje

Uporabniki še vedno prejemajo programsko opremo, ki ni temeljito stestirana (pravimo, da ima preveč uši ali hroščev). To povzroča pri uporabnikih nepotrebne frustracije in stroške, kar je prava nesreča. Vendar je še bolj presenetljivo dejstvo to, da programerji, ki so napisali program, dostikrat ne poznajo natančne poti, ki bi jim pokazala, kaj je bilo narobe, če je program odpovedal. Testiranje in odpravljanje programskih napak še vedno v pretežni meri uporablja metodo tipanja ("trial and error").

Glede na to koliko nejevolje, razočaranj in stroškov povzroča premalo preverjena programska oprema tako uporabnikom kot tudi razvojnemu osebju, je za računalniško znanost že pravi škandal (tako H. Lieberman), da je ta problem v tolikšni meri zanemarljiva. Sodobna programska orodja kot pomoč pri testiranju so le malo boljše kot tista izpred 30 let. Žalostno je ugotoviti, da še dandanašnji veliko programerjev navaja kot svojo tehniko testiranja ("debugging") "vstavljanje ukazov za tiskanje".

In vendar obstaja že veliko boljših načinov oz. metod, kako uspešno stestirati program. Vemo, da so programi kompleksni izdelki. Zakaj ne bi vključili računalnika tudi pri testiranju? Računalnik bo zlasti odigral aktivno vlogo pri razgrajevanju kompleksnosti. Računalniki so sedaj hitri, z velikim osrednjim pomnilnikom in velikimi zmogljivostmi na diskih. Zakaj ne bi uporabili teh prednosti, da bi programer prišel čimprej do informacij, ki mu bodo pomagale razumeti delovanje programa. Zakaj ne bi uporabili računalniške grafike, ki bi programerju predočila vedenje programa.

Veliko strokovnjakov dvomi v možnosti večjih izboljšav pri procesu testiranja programov. Menijo, da je to pač trdo delo. Mnogi programerji odklanjajo kakršnokoli orodje pri testiranju, češ dobri programerji tega ne potrebujejo. Resnica je drugačna. Dobra orodja lahko navidezno nedosegljivo napako (uš) prikažejo še tako izkušenemu programerju kot običajno in kot tako jo bo zlahka odpravil.

So tudi ljudje, ki zaupajo le formalnim tehnikam kot je programska verifikacija. Verjamejo, da tovrstne tehnike odpravljajo potrebo po testiranju. Vemo pa, da je nerealno

pričakovati popolno odpravo napak, čeprav se programska oprema razvija kar naprej, oziroma jo stalno dopolnjujejo. Orodja za testiranje so dobrodošla tudi pri dopolnjevanju programov.

Na podlagi ankete po internetu, ki jo je izvedel M. Eisenstadt v letih 1992 - 1993, je bilo ugotovljeno, da povzročajo več kot 50 % težav:

- velike časovne ali prostorske razlike med osnovnim vzrokom in začetnim simptomom
- napake, pri katerih testna orodja niso uporabna.

Anketa je tudi pokazala, da je skoraj polovico pomanjkljivosti (uši) povzročala dobaviteljeva strojna ali programska oprema. Nedavno smo v RRC dobili od enega uporabnika pisno sporočilo, da neki naši programi ne delujejo več pravilno. Analiza programov na drugi strojni opremi pa je pokazala nasprotno. Poklicali so vzdrževalce strojne opreme, ki so porabili več kot 40 ur, da so našli napako in sicer na strojni opremi (začeli so jo seveda spet iskati v programih). Na dobavo strojne opreme RRC seveda ne more vplivati. Kljub temu so še sedaj nekateri ljudje pri uporabniku prepričani, da je bila krivda v RRC, ker preprosto ne morejo verjeti, da lahko pride do napak v delovanju tudi zaradi dobaviteljeve opreme.

Kdor hoče program dobro stestirati, ga mora znati pregledati z različnih vidikov. Da bi omogočili večjo berljivost in razumevanje programov, so razvili različne tehnike. Med najbolj učinkovite gotovo sodi vizualizacija izvorne programske kode. Sicer pa je na boljše razumevanje programov opozoril D. Knuth v svojem članku "Literate programming" (*Comput. J.* 27,2,1984), kjer apelira na avtorje programov, naj začno gledati na svoje izdelke kot literarne in naj jih dokumentirajo na tak način, da jih bodo ljudje razumeli.

In podatki?

V nekaterih oddelkih za informatiko večjih družb pogosto naletimo na prepričanje, da so z uspešnim prevzemnim testom rešili vse probleme v zvezi s predvideno aplikacijo. Iz-

kušnje pa nas učijo, da povzročča uporaba vsakega orodja večje ali manjše težave. To velja še zlasti za programe, ki obravnavajo večje količine podatkov. Pri tem ne smemo pozabiti na kakovost podatkov.

Novjše študije (D.M. Strong, Y.W. Lee and R.Y. Yang: Data Quality in Context, Comm. of ACM, May 1997 / Vol.40, No. 5) v zvezi s kakovostjo podatkov namreč kažejo, da tovrstni problemi naraščajo. Slaba kakovost podatkov (KP) v podatkovnih bazah je draga tako v socialnem kot tudi v ekonomskem smislu. Organizacijske podatkovne baze (PB) so shranjene v širšem kontekstu informacijskih sistemov (IS). V tem širšem kontekstu zbirajo podatke iz večjih podatkovnih virov in jih shranjujejo v PB. Iz tako zbranih podatkov oblikujejo informacije za sprejemanje poslovnih odločitev.

V tem širšem kontekstu IS lahko nastanejo kjerkoli problemi kakovosti podatkov, čeprav nekateri smatrajo KP kot naravno lastnost podatkov, ki naj ne bi bila odvisna od proizvodnje in uporabe podatkov. V literaturi o kakovosti pa velja pravilo, da kakovosti ne smemo ocenjevati neodvisno od uporabnikov. Podobno pravilo velja za KP, ki jo je treba ocenjevati preko okvirov naravnih lastnosti (ne moremo torej gledati le shranjenih podatkov, ampak je treba vključiti tudi podatke v proizvodnih in uporabniških procesih) in pri tem upoštevati ljudi - uporabnike podatkov. Njihova ocena KP postaja vse pomembnejša, ker imajo vedno boljše kontrolo nad računalniškim okoljem in predvsem nad podatki, ki jih uporabljajo.

Da bi lahko kakovost podatkov bolje opredelili, jo razvrščajo v več kategorij:


- naravne lastnosti KP (točnost, objektivnost, možnost, uveljavljenost)
 - dostopnost KP (dostopnost, varnost dostopa)
 - povezanost (kontekstualnost) KP (relevantnost, dodana vrednost, pravočasnost, popolnost, količina podatkov).
- Običajne tehnike kontrole (n.pr. preverjanje, omejitve integritete PB, programska kontrola ažuriranja PB), ki zagotavljajo KP, so znatno izboljšale točnost podatkov. Vendar to ne

zadošča, kakor tudi ne sama kontrola nad shranjevanjem podatkov. Vključiti je treba revizijske tehnike, ki lahko nadzirajo procese, kjer nastajajo podatki.

Tudi dostopnost KP se razlikuje v tehničnem oz. uporabniškem smislu. Konvencionalni pristop obravnava dostopnost kot tehnično, računalniško zadevo. To pomeni, da so podatkovni skrbniki poskrbeli za dostop, če so strežniki povezani z uporabniki in delujejo, če ima uporabnik dovoljenje za uporabo podatkov itn. Uporabniki podatkov hočejo več, namreč enostavno manipulacijo s podatki, da zadostijo svojim potrebam. Tako so podatki iz posameznih avtonomnih sistemov tehnično dostopni, za uporabnike pa niso dostopni, ker so vrednosti enakih podatkovnih postavk različno definirane ali predstavljene. Velike količine podatkov so lahko tehnično dostopne, ne pa za uporabnike, ker so dostopni časi predolgi. Informatiki morajo razlikovati med tehnično dostopnostjo do podatkov, kar je njihova naloga, in med širšim pojmom dostopnosti, kakor jo pojmujejo uporabniki. Čim smo dojeli to razliko, lahko uporabimo druge tehnologije, n.pr. podatkovno skladiščenje, ki lahko oskrbi uporabnika z manjšo količino bolj relevantnih podatkov.

Uporabniki podatkov vrednotijo KP relativno glede na svoje naloge. Pri tem se značilnosti KP lahko s časom spreminjajo. Zato je razumljiva zahteva, da morajo zasnove zagotavljati prožne sisteme, ki vzpostavljajo kakovostne podatke (n.pr. enostavna manipulacija in združevanje podatkov), ker je sicer alternativa stalno vzdrževanje podatkov in sistemov.

Ugotovili smo, da je kakovost podatkov skrb tako informatikov kot tudi uporabnikov. Informatiki so dolžni zagotoviti tehnične pogoje, uporabniki pa morajo dobro poznati druge lastnosti KP, da ne prihaja do okvar podatkovnih baz. Kadar pa kreirajo ali ažurirajo PB pretežno s pomočjo javnih evidenc, je kakovost podatkov še posebej pomembna. Zato je nujno potrebno sodelovanje med informatiki in uporabniki, če hočejo proces zagotavljanja kakovosti čim bolj avtomatizirati.

Svoj izvod dobite na naslovu:

Kolap\$, Kardeljeva ploščad 17, 1000 Ljubljana
kolaps@uni-lj.si
<http://www.pasef.ef.uni-lj.si/kolaps/>

Iz zadnje številke:

Dosje: ZATON NACIONALNE DRŽAVE

dr. Frane Adam: NACIONALNO V KONTEKSTU GLOBALIZACIJE
intervju: Milan Kučan, dr. Edmund Farhat in NSK

Tadej Mali: EKSPERIMENT ZA NOVO TISOČLETJE

dr. Ivan Ribnikar vs. Bernard Brščič: DRUGI STEBER IN PRIVATIZACIJSKA LUKNJA

Aljoša Valentinčič: DOGODKOVNE ŠTUDIJE

Matjaž Jager: NEKAJ O ENAKOSTI MED LJUDMI

Monika Osvald: ANDY WARHOL IN OSEMDESETA

Mark Sagoff: ŽIVALI KOT IZNAJDBE

Kolaps
ECONOMIA - PHILOSOPHIA - IUS - SCIENTIA