

## **Da li je potrebno testiranje softvera?**

Predrag Vujić

Matematički fakultet, Studentski trg 16  
11000 Belgrade, Serbia  
wujic88@gmail.com

## Sadržaj

|     |                                                |   |
|-----|------------------------------------------------|---|
| 1   | Uvod .....                                     | 3 |
| 2   | Profesionalni kodeks (IEEE-CS/ACM) .....       | 4 |
| 2.1 | Javni interes .....                            | 4 |
| 2.2 | Klijent i poslodavac .....                     | 5 |
| 2.3 | Proizvod .....                                 | 5 |
| 2.4 | Rasuđivanje .....                              | 5 |
| 2.5 | Upravljanje .....                              | 6 |
| 2.6 | Profesija .....                                | 6 |
| 2.7 | Kolegijalnost .....                            | 6 |
| 2.8 | Odnos prema sebi .....                         | 6 |
| 3   | Slučaj masovnog pada telefonskog sistema ..... | 7 |
| 4   | Analiza slučaja .....                          | 7 |
| 4.1 | Javni interes .....                            | 7 |
| 4.2 | Klijent i poslodavac .....                     | 7 |
| 4.3 | Proizvod .....                                 | 8 |
| 4.4 | Upravljanje .....                              | 8 |
| 4.5 | Profesija .....                                | 8 |
| 5   | Zaključak .....                                | 9 |

## 1. Uvod

„Informacione tehnologije“ je savremeni izraz koji označava kombinaciju računarskih i telekomunikacionih tehnologija koje se upotrebljavaju u prikupljanju, obradi, skladištenju, razmeni i izdavanju informacija. Zavisno od konteksta, u informacione tehnologije se mogu ubrajati i druge tehnologije za rukovanje informacijama, kao npr. tehnologije za proizvodnju ili skladištenje papirnih dokumenata.

Imaju centralnu i rastuću ulogu u trgovini, industriji, nauci, edukaciji, politici, medicini, zabavi i društvu uopšte. Osobe koje učestvuju u razvoju informacionih sistema ujedno doprinose i razvoju informacionih tehnologija. U okviru informacionog sistema u te osobe spadaju projektanti, razvijaoči, istraživači, analitičari, dizajneri, testeri i održavaoci informacionog sistema.

Prilikom razvoja softvera, uglavnom se najviše pažnje posvećuje analizi, projektovanju i analizi datog softvera. Sa druge strane se često zapostavlja njegovo testiranje i održavanje. Postoji više razloga zbog toga, kao što je kratak rok, nedovoljan budžet, nedostatak kadrova, loše vođenje projekta ili kombinacija svega navedenog.

Testiranje softvera se sastoji iz dinamičke verifikacije ponašanja softvera na konačnom skupu testiranja, pogodno selektovanog iz uglavnom beskonačnih područja izvršavanja u odnosu na očekivano ponašanje. Predstavlja dodatni napor. Zahteva produžavanje roka izrade softvera, kao i povećavanje troškova izrade. Sa druge strane, testiranje softvera smanjuje mogućnost štetnog uticaja na rad kompletnog sistema i tako se sprečava pojava nepredviđenih problema, koji mogu da izazovu dodatne nepredviđene troškove i produžavanje roka izrade ili mogu pak biti fatalne po ceo sistem.

U ovom radu biće opisan slučaj puštanja u rad softvera koji nije testiran, kroz primer masovnog pada telefonskog sistema, kao i posledica koje je izazvao. Obuhvaćen je i profesionalan kodeks čiji je cilj da pospeši kvalitet izrade softvera i na osnovu koga je moguće utvrditi buduće nepravilnosti u radu.

## 2. Profesionalni kodeks (IEEE-CS/ACM)

Profesionalni kodeksi su nastali iz potrebe razrešavanja određenih problema koji se odnose na etičke principe, međusobne odnose između razvijaoa softvera, odnose razvijaoa i samog razvoja softvera, odnosa razvijaoa i naručioca softvera, kao i uticaja samog softvera na njegove krajnje korisnike i javnost uopšte.

Napisani su od strane osoba i organizacija koje su istaknute u oblasti kojom se bave. Profesionalni kodeksi su dokumenti sa skupom formalnih principa koji treba da predstavljaju temelj svakog softverskog projekta i glavna smernica svim učesnicima u izradi softvera. Ukoliko se pridržava profesionalnim kodeksima, smanjuje se mogućnost nastajanja problema prilikom razvoja i korišćenja softvera.

IEEE-CS/ACM (The International Standard for Professional Software Development and Ethical Responsibility) je standard koji predstavlja etički kodeks i profesionalnu praksu u softverskom inženjerstvu. Proizvod je zajedničkog rada organizacija IEEE-CS (IEEE Computer Society) i ACM (Association for Computing Machinery).

Sastoji se od osam glavnih principa:

1. Javni interes
2. Klijent i poslodavac
3. Proizvod
4. Rasuđivanje
5. Upravljanje
6. Profesija
7. Kolegijalnost
8. Odnos prema sebi

### 2.1. Javni interes

Softverski inženjeri deluju u skladu sa javnim interesom. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Odobriti softver uz opravdano uverenje da je bezbedan, da zadovoljava specifikacije, da je prošao testove i da ne ugrožava kvalitet života, privatnost ili šteti okolini.
2. Obavestiti korisnike softvera o svim potencijalnim opasnostima.
3. Preuzeti puno odgovornost za svoj rad.
4. Uskladiti interese softverskog inženjeram poslodavca, klijenta i korisnika sa javnim interesom.
5. Pokazati dobru volju da svoja stručna znanja stave na raspolaganje u dobrotvorne svrhe i pridonose obrazovanju javnosti o svojoj struci.

## 2.2. Klijent i poslodavac

Softverski inženjeri deluju u skladu sa interesima njihovog klijenta ili poslodavca, a koji nisu u suprotnosti sa javnim interesom. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Očuvati privatnost svih poverljivih informacija dobijenih u njihovom profesionalnom radu, uz uslov da je ta poverljivost u skladu sa javnim interesom i zakonom.
2. Pružati usluge u svojim područjima stručnosti i pošteno i otvoreno priznati granice svog iskustva i obrazovanja.
3. Osigurati da je svaki dokument na koji se oslanjaju odobren, kada je to potrebno, od ovlašćene osobe.

## 2.3. Proizvod

Softverski inženjeri se obavezuju da će obezbediti da njihovi proizvodi i njegove prateće izmene zadovoljavaju najviše moguće standarde. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Težiti visokom kvalitetu, prihvatljivoj ceni i razumnim rokovima, osiguravajući da su svi značajni kompromisi jasni i prihvaćeni od poslodavca i klijenta.
2. Osigurati realni smisao i dohvatljive ciljeve svakog projekta na kojem rade ili ga predlažu.
3. Obezbediti odgovarajuće testiranje, otklanjanje grešaka i proveru softverskog proizvoda i pratećih dokumenata na kojima rade.
4. Odnositi se prema svim oblicima održavanja softvera sa istim profesionalizmom kao prema razvoju novog.

## 2.4. Rasuđivanje

Softverski inženjeri održavaju integritet i nezavisnost u svojim stručnim rasuđivanjima. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Obavestiti sve zainteresovane stranke o onim suprotstavljenim interesima koji se ne mogu izbeći na razuman način.
2. Ne učestvovati u finansijskim prevarama kao što je mito, dupla naplata ili druge neprikladne finansijske radnje.
3. Overiti samo dokumasta sa kojima se slažu, a pripremljeni su bilo pod njihovim nadzorom bilo unutar njihovog područja stručnosti.
4. Ograničiti sve tehničke procene potrebom za podrškom i održavanjem ljudskih vrednosti.

## 2.5. Upravljanje

Menadžeri i lideri softverskog inženjerstva se pridržavaju i podstiču etički pristup u upravljanju razvoja softvera i održavanju. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Obezbediti da softverski inženjeri budu obavešteni o standardima pre nego što ih trebaju primeniti.
2. Osigurati dobro vođenje projekta na kojima rade.
3. Podeliti radne zadatke uzimajući u obzir stečeno radno iskustvo i potrebno obrazovanje, kao i želju za daljim unapređivanjem iskustva i obrazovanja.
4. Ne kazniti nikoga za izražavanje etičkih briga o projektu.

## 2.6. Profesija

Softverski inženjeri moraju da unapređuju ugled i integritet profesije u skladu sa javnim interesom. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Podržavati javno razumevanje softverskog inženjerstva.
2. Tačno navesti osobine softverskog proizvoda na kome rade.
3. Preuzeti odgovornost za otkrivanje, ispravljanje i prijavljivanje grešaka u softverskom proizvodu uključujući i dokumente na kojima rade.

## 2.7. Kolegijalnost

Softverski inženjeri se međusobno podržavati i ophode pošteno. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Pomoći kolegama u stručnom usavršavanju.
2. Nepriistrano saslušati mišljenja, brige ili pritužbe kolega.
3. U područjima koja prevazilaze njihovu vlastitu stručnost, savetovati se sa drugim stručnjacima kompetentnim za to područje.

## 2.8. Odnos prema sebi

Softverski inženjeri se kontinuirano stručno usavršavaju i podstiču etički pristup radu u svojoj profesiji. Posebno, u skladu sa prilikama, softverski inženjeri se obavezuju da će:

1. Unaprediti svoje znanje o razvoju na području analize, specifikacije, projektovanja, razvoja, održavanja i testiranja softverskih proizvoda i prateće dokumentacije, kao i o upravljanju razvojnim procesima.
2. Unaprediti svoje razumevanje programskih proizvoda i prateće dokumentacije kojima se služe, te okoline u kojoj će biti upotrebljeni.
3. Unaprediti svoju sposobnost stvaranja sigurnih, pouzdanih i korisnih kvalitetnih softverskih proizvoda po prihvatljivoj ceni i u razumnim vremenskim okvirima.
4. Unaprediti svoju sposobnost pisanja tačne, sadržajne i dobro oblikovane dokumentacije.

### 3. Slučaj masovnog pada telefonskog sistema

U nekoliko većih gradova u Sjedinjenim Američkim Državama desio se slučaj masovnog pada telefonskog sistema 1991. godine na osam sati. Pad su prouzrokovale tri pogrešne kompjuterske instrukcije, koje su bile deo nove verzije softvera koji je koristio telefonski sistem. Cilj nove verzije je bio poboljšanje performansi opreme. Izmene u verziji softvera su smatrane isuviše male da bi se on testirao, pa su rukovodioci odlučili da novu verziju softvera puste u rad bez standardne procedure testiranja.

### 4. Analiza slučaja

Slučaj masovnog pada telefonskog sistema je u to vreme samo potvrdio i dodatno povećao strahovanja da će rast korišćenja softvera i povećanje njegove kompleksnosti sve više biti osetljivo na njegove padove i propuste. Ovaj slučaj, kao i slučajevi slični ovom su naterali društvo na pronalaženje rešenja koja bi mogla da spreče ili pak da ublaže posledice ovakvih padova. Jedno od tih rešenja su profesionalni kodeksi. Kodeks IEEE-CS /ACM je nastao 1997. godine, šest godina posle pada telefonskog sistema. Da je postojao u to vreme, ili neki sličan kodeks, možda nebi ni došlo do ovakvog scenarija. Ovakvi scenariji su direktno podstakli nastajanje profesionalnih kodeksa.

Ukoliko poredimo ovaj slučaj sa profesionalnim kodeksom IEEE-CS/ACM, vidimo da je prekršeno pet od osam principa kodeksa i to sledeći principi:

1. Javni interes
2. Klijent i poslodavac
3. Proizvod
4. Upravljanje
5. Profesija

#### 4.1. Javni interes

Padom telefonskog sistema bila je onemogućena telefonska komunikacija između ljudi na osam sati. Nije lako odrediti veličinu posledice tog pada, počevši od elementarnih potreba kao što su hitna pomoć, vaspitnici, policija i službe za informacije, a zatim i kompanije za čije poslovanje telefonska komunikacija igra veliku ulogu. Njihov rad i funkcionisanje je bilo onemogućeno i ljudi koji su zavisili od rada tih organizacija su bili nemoćni, i oni predstavljaju krajnje žrtve pada sistema. Javni interes je tada bio apsolutno narušen. Softver nije bio bezbedan i pouzdan, nije prošao potrebne testove i ugrozio je kvalitet života njegovih korisnika. Klijent nije bio obavešten o mogućim posledicama puštanja u rad netestiranog softvera.

#### 4.2. Klijent i poslodavac

Klijent je delovao u skladu sa interesima poslodavca, jer je radio na unapređenju verzije softvera, međutim puštanjem u rad netestiranog softvera i njegovim padom, klijent nije delovao u skladu sa javnim interesom. Narušavanje ovog principa je direktna posledica narušavanja javnog interesa.

#### **4.3. Proizvod**

Očigledno je da nova verzija softvera nije bila ni blizu najviših profesionalnih standarda na koji se softverki inženjeri obavezuju prilikom isporučivanja i modifikacije softvera.

#### **4.4. Upravljanje**

Donošenjem odluke da se softver pusti u rad bez bilo kakvog testiranja, čak i ako su u pitanju minimalne modifikacije, je izazvalo fatalne posledice. Osoba koja je donela odluku napravila je lošu procenu i nije uzela u obzir etički pristup upravljanju razvojem softvera i održavanju, i nije razmotrila moguće posledice pada softvera.

#### **4.5. Profesija**

Kako je javnost direktno trpela posledice pada telefonskog sistema, ugled i integritet softverskog inženjerstva su dovedeni u pitanje. Na taj način je ugrožena sama profesija softverskog inženjerstva.

## 5. Zaključak

Ne postoji savršen softver, kao ni softver koji je završen 100% do kraja. Uvek postoji mesto za uočavanje i otklanjanje grešaka, popularno nazvanih bug-ova, i uvek ima mesta za modifikacije i unapređivanje softvera i time potencijalno stvaranje novih grešaka. Na pitanje da li testirati softver ili ne, odgovor je jednostavan - Testirati.

Sledeće je pitanje kada treba testirati softver? To pitanje zahteva malo duži odgovor. Ranije se sa testiranjem softvera počinjalo tek po završetku pisanja koda. Novije metodologije razvoja softvera kao što su Agilne metodologije, uključuju testiranje već i tokom pisanja samog koda od strane razvijaoca a i po završetku pisanja koda , od strane testera. Agilne metodologije su zasnovane na inkrementalnom pristupu razvoja softvera, što znači da se softver razvija kroz evolutivne cikluse. Iz tog razloga je neophodno povećati mere kvaliteta softvera jer se jednom napisani kod često nadgrađuje novim funkcionalnostima u Agilnim metodologijama, i takav kod je sklon anomalijama.

Koliko je potrebno testirati softver? Odgovor je, zavisi. Od čega zavisi? Zavisi od više stvari: Veličine softverskog proizvoda, broja korisnika tog softvera, procene troškova i štete u slučaju pada, brzine povratka na stabilnu verziju i ispravljanje grešaka.

U opisanom slučaju očigledno je da je netestiranje softvera dovelo do fatalnih posledica. Međutim, opisani slučaj predstavlja pad informacionih sistema telefonske industrije nekoliko većih gradova u Sjedinjenim Američkim Državama. Korisnici tog sistema broje se u milionima. Svaki minut prestanka rada kompletnog sistema broji se u milionima dolara gubitaka. U ovakvim slučajevima testiranje softvera ne treba dovoditi u pitanje. Obavezno testirati, ma koliko to bilo skupo i ma koliko to dugo trajalo.

U situacijama kada okolnosti nisu toliko drastične kao što je softver telefonske industrije, često se dolazi u situaciju da se softverski proizvod iz raznoraznih razloga pusti u rad sa ne puno ali ne i bez testiranja. U takvim situacijama, što i sam kodeks nalaže, klijent mora biti svestan mogućih nedostataka i mora imati spremnu strategiju oporavka, kako bi se šteta svela na minimum i kako bi korisnici ostali što manje pogođeni bilo kakvim negativnim scenarijom.

## Reference

1. ACM Code of Ethics and Professional Conduct, available at <http://www.acm.org/about/code-of-ethics>
2. Code of conduct for BCS members, <http://www.bcs.org/upload/pdf/conduct.pdf> [Online; accessed 15-Oct-2013]
3. The international standard for professional software development and ethical responsibility
4. Wikipedia Software testing, [Online]. Available: [http://en.wikipedia.org/wiki/Software\\_testing](http://en.wikipedia.org/wiki/Software_testing)
5. Zubairi, J.A.: To Test or Not to Test the Software: A Case Study on Ethics in Computing, mathematics and Computer Science SUNY at Fredonia