

Vrsta rada: Originalni naučni rad

Primljen: 10.06.2023.

Prihvaćen: 03.07.2023.

UDK: 004.415.53

004.78:004.85

Revolucionizacija softverskog testiranja: Uticaj veštačke inteligencije, mašinskog učenja i Interneta stvari

Vukašin Jeremić,¹ Rocsana Bucea-Manea-Țonis,² Slavimir Vesić³ i Hana Stefanović⁴

¹ JV Solutions; vjeremic90@gmail.com

² Nacionalni univerzitet fizičkog vaspitanja i sporta, doktorski program, Bukurešt, 060057, Rumunija; rocsense39@yahoo.com

⁴ Visoka škola strukovnih studija za informacione tehnologije – ITS, Beograd; hana.stefanovic@its.edu.rs

³ Visoka škola strukovnih studija za informacione tehnologije – ITS, Beograd; slavimir.vesic@its.edu.rs

Apstrakt

Testiranje softvera predstavlja jedan od najvažnijih aspekata životnog ciklusa i razvoja softvera, budući da garantuje funkcionalnost i prihvatljivost softverskih aplikacija za korisnike. Rad pruža detaljan pregled testiranja softvera, naglašavajući njegov značaj, različite metodologije i izazove, ali i uticaj novih tehnologija poput veštačke inteligencije (AI), mašinskog učenja (ML) i Interneta stvari (IoT). Istaknuta je i uloga tekućih istraživanja u unapređenju softverskog testiranja kako bi se poboljšali kvalitet i pouzdanost softvera.

Testiranje softvera, u koje spada širok spektar aktivnosti tokom životnog ciklusa i razvoja softvera, ključno je za procenjivanje funkcionalnih i nefunkcionalnih osobina softvera, čime se olakšavaju identifikacija i otklanjanje operativnih nedostataka. Sve veća složenost i raznovrsnost softvera predstavljaju značajan izazov za razvoj efikasnih strategija i alata za testiranje. Ovaj rad se bavi perspektivnim tehnologijama za testiranje softvera i zalazi u detalje fundamentalnih principa softverskog testiranja, istražujući inovativne metodologije za kreiranje test slučajeva i objašnjavajući goruća pitanja u okviru samog procesa testiranja i njegove uloge unutar savremenih razvojnih protokola. Autori se takođe bave testerskim rešenjima koja su prilagođena jedinstvenim zahtevima brzorastućeg tržišta aplikacija.

Cljučne reči: testiranje softvera, veštačka inteligencija, mašinsko učenje, Internet stvari, automatizacija testiranja

Uvod

Testiranje softvera je jedan od najvažnijih procesa za procenu funkcionalnosti softverskih proizvoda, dizajniran sa ciljem otkrivanja grešaka i nedostataka. Testiranje je neophodno kao potvrda da softver zaista ispunjava očekivanja kupaca, da bi se osigurala njegova pouzdanost, očuvao kredibilitet, unapredilo korisničko iskustvo i ublažili potencijalni rizici u vezi sa neispravnim softverom (Pece i Jang, 2007). Ako izuzmemo njegov značaj, testiranje softvera sa sobom nosi izazove koji zahtevaju inovativna rešenja i istraživanje (Bajzer, 1990).

U suštini, testiranje softvera je proces kojim se potvrđuje da je krajnji rezultat u skladu sa očekivanim ishodima. Podrazumeva ocenjivanje softvera kako bi se otkrile i ispravile sve greške ili nedostaci koji mogu da utiču na kvalitet. Inženjeri kvaliteta, odnosno tester, moraju biti u stanju da odrede prioritete svojih zadataka u skladu sa ozbiljnošću svake pojedinačne greške ili nedostatka.

Testiranje softvera prevazilazi prostu identifikaciju grešaka ili nedostataka. Ono takođe doprinosi ukupnom poboljšanju kvaliteta softvera i predstavlja isplativo rešenje koje dovodi do poboljšanja proizvoda u smislu funkcionalnosti, preciznosti i bezbednosti podataka. Ovo je proces koji se ponavlja i traje čak i nakon što je proizvod kompletiran.

Glavni cilj testiranja softvera jeste verifikacija i validacija da je softver u potpunosti kompletiran i da ispunjava sve tehničke zahteve. Tester je dužan da blagovremeno prijavi sve tehničke probleme i da se postara da se sve greške i kvarovi uklone pre puštanja softvera na tržište.

Testerima imaju zadatak da kreiraju kvalitetne test slučajeve i precizne izveštaje o problemima. Pravilno testiranje softvera organizacijama značajno štedi vreme, trud i novac neophodan za razvoj i prodaju softverskih proizvoda. Ono takođe pomaže u optimizaciji, jer garantuje da će aplikacija ili proizvod optimalno raditi u svim traženim uslovima i na svim operativnim sistemima i internet pretraživačima.

Ovako strogo testiranje dovodi do unapređenog korisničkog iskustva i povećava zadovoljstvo kupaca, što na kraju dovodi do povećanog profita za organizaciju. Testiranje softvera je osmišljeno kako bi se osiguralo da softver koji izađe na tržište bude stabilan i da radi kako je predviđeno.

Ključno je prepoznati značaj testiranja softvera, jer softverske greške i nedostaci mogu da dovedu do izuzetno skupih i opasnih posledica. Proces testiranja predstavlja kreativan i intelektualno zahtevan zadatak za testere i zahteva poznavanje različitih faktora i principa, poput skalabilnosti, upotrebljivosti i bezbednosti (Divjani, 2020).

Metode

Testiranje softvera se sprovodi korišćenjem različitih metodologija, od kojih svaka ima jedinstvene prednosti i mane. U ove metodologije spadaju: jedinično, integracijsko i sistemsko testiranje, kao i testiranje prihvatljivosti od strane korisnika. Jedinično testiranje se odnosi na nezavisno testiranje pojedinačnih softverskih komponenti kako bi se potvrdila njihova ispravnost (Majers i saradnici, 2011). Integracijsko testiranje kombinuje individualne jedinice i zajedno ih testira kako bi se utvrdili problemi u integraciji (Peri, 2006). Sistemsko testiranje podrazumeva testiranje kompletnog softvera kako bi se utvrdilo da li ispunjava određene zahteve i da li se ponaša kako bi trebalo (Fjoster i Grejem, 1999). Poslednji tip jeste testiranje prihvatljivosti, kojim se proverava da li softver ispunjava kriterijume prihvatljivosti i da li je spreman za korišćenje (Kejner i saradnici, 1993).

Pregled literature

Oblast softverskog testiranja je obimna, sa ogromnom količinom literature koja se bavi značajem, metodologijama, izazovima i uticajem novih tehnologija na testiranje softvera.

Bajzer (1990) je među prvima napisao značajan rad o tehnikama softverskog testiranja, nudeći pregled različitih metoda testiranja i njihovih primena. Njegov rad postavio je temelje za mnoga kasnija istraživanja softverskog testiranja. Majers, Sandler i Badžet (2011) takođe su ponudili detaljno istraživanje softverskog testiranja u svojoj knjizi „Umetnost softverskog testiranja“, gde objašnjavaju različite nivoe testiranja softvera, uključujući integracijsko i sistemsko testiranje, kao i testiranje prihvatljivosti.

Pece i Jang (2007) ističu značaj testiranja u osiguravanju pouzdanosti softvera i zadovoljstva korisnika. Oni su prepoznali da je testiranje softvera ključna procedura u životnom ciklusu i razvoju softvera kako bi se otkrile greške i nedostaci koji mogu da utiču na performanse.

Kejner, Falk i Ngijen (1993) raspravljaju o izazovima povezanim sa testiranjem softvera. Oni ističu nekoliko problema kao što su, na primer, izbor odgovarajuće metodologije testiranja, potom obezbeđivanje dovoljne pokrivenosti testom, rešavanje nejasnih ili nepotpunih zahteva i upravljanje vremenskim i budžetskim ograničenjima.

Kada je reč o novim tehnologijama, Džang i saradnici (2020) raspravljaju o potencijalima veštačke inteligencije i mašinskog učenja u transformaciji softverskog testiranja. Oni sugerišu da bi navedene tehnologije mogle da automatizuju kreiranje testova, poboljšaju detekciju grešaka, predvide kvalitet softvera i pruže dragoceni uvid u sam proces testiranja. Slično tome, Stol i saradnici (2016) ističu izazove i mogućnosti povezane sa IoT uređajima, primećujući da povećana složenost i obim testiranja softvera zahtevaju inovativne tehnologije i alate za testiranje.

Zaključak koji se nameće jeste da postoji obimna literatura o testiranju softvera koja detaljno objašnjava njegov značaj, različite metodologije koje se koriste, izazove sa kojima se testerima susreću, ali i uticaj novih tehnologija. Istraživači se slažu da su kontinuirani razvoj i istraživanje softverskog testiranja neophodni kako bi se prevazišli izazovi i iskoristile mogućnosti koje pružaju nove tehnologije.

Rezultati

Sve veća složenost softverskih aplikacija i rapidan razvoj novih tehnologija, poput veštačke inteligencije, mašinskog učenja i Interneta stvari, značajno su izmenili oblast softverskog testiranja. Veštačka inteligencija i mašinsko učenje pokazuju potencijal da revolucionišu testiranje softvera zahvaljujući automatskom generisanju testova, unapređenim sposobnostima za otkrivanje grešaka, mogućnosti da predvide kvalitet softvera i mogućnosti da pruže uvid u sam proces testiranja (Džang i saradnici, 2020). Štaviše, zahvaljujući povećanom broju i popularnosti IoT uređaja, složenost i obim softverskog testiranja su značajno porasli i sada zahtevaju inovativne tehnike i alate za testiranje koji su u stanju da adekvatno odgovore na složene izazove povezane za IoT sistemima (Stol i saradnici, 2016).

Neprekidni razvoj i povećana složenost softverskih aplikacija, zajedno sa rapidnim razvojem novih tehnologija poput veštačke inteligencije, mašinskog učenja i Interneta stvari, značajno su uticali na oblast softverskog testiranja. Potencijal veštačke inteligencije i mašinskog učenja da automatizuju proces generisanja testova istinski su revolucionisali testiranje softvera (Džang i saradnici, 2020).

Pored toga, povećan broj i popularnost IoT uređaja drastično su povećali složenost i obim softverskog testiranja, zahtevajući inovativne tehnike i alate za testiranje koji su u stanju da adekvatno odgovore na složene izazove u vezi sa IoT sistemima (Stol i saradnici, 2016).

Digitalno doba dovelo je do toga da softver bude prisutan u svakom aspektu našeg života, zbog čega su besprekoran rad i performanse postali imperativ (Jorgensen, 2016). Upravo zato testiranje softvera ne samo da osigurava pouzdanost softvera nego i sprečava potencijalni gubitak korisnika izazvan defektnim aplikacijama.

Svaka od metodologija softverskog testiranja, kao što su jedinično, integracijsko i sistemsko testiranje, ali i testiranje prihvatljivosti, igra ključnu ulogu u životnom ciklusu i razvoju softvera. Svaka od navedenih metodologija obezbeđuje različite aspekte softverskog proizvoda, poput funkcionalnosti, pouzdanosti i prihvatanja od strane korisnika (Majers i saradnici, 2011; Peri, 2006; Fjuster i Grejem, 1999; Kejner i saradnici, 1993).

Međutim, uprkos svom značaju, testiranje softvera donosi i različite izazove, kao što su prepoznavanje odgovarajuće metodologije testiranja, obezbeđivanje dovoljne pokrivenosti testom, rešavanje dvosmislenih ili nepotpunih zahteva i upravljanje vremenskim i budžetskim ograničenjima (Kejner i saradnici, 1999). Kako softverski sistemi postaju sve kompleksniji i zavisniji jedni od drugih, tako efikasne metodologije za testiranje postaju sve neophodnije, što zahteva kontinuirano istraživanje i razvoj u ovoj oblasti.

Integracija veštačke inteligencije i mašinskog učenja u softversko testiranje može značajno da poveća efikasnost samog testiranja. Algoritmi mašinskog učenja, na primer, mogu da analiziraju istorijske podatke i predvide oblasti podložne greškama, dok veštačka inteligencija automatizuje svakodnevne zadatke, smanjujući prostor za ljudske greške i omogućava testerima da se fokusiraju na složenije zadatke (Džang i saradnici, 2020).

Rast popularnosti IoT uređaja takođe je doprineo sve većoj složenosti softverskog testiranja. Budući da Internet stvari predstavlja jedinstvenu kombinaciju hardvera i softvera, on zahteva testiranje u različitim uslovima, odnosno inovativne tehnike i alate za testiranje koji su u stanju da se nose sa povećanom složenošću i raznovrsnošću IoT sistema (Stol i saradnici, 2016).

Budućnost softverskog testiranja

Oblast softverskog testiranja nastaviće da se menja i razvija i u budućnosti pod uticajem sve veće složenosti softverskih aplikacija i rapidnog razvoja tehnologija, u koje spadaju veštačka inteligencija, mašinsko učenje i Internet stvari. Ove nove tehnologije poseduju potencijal da automatizuju i unaprede različite aspekte softverskog testiranja, što će neizbežno dovesti do pouzdanijih i kvalitetnijih softverskih proizvoda. Međutim, pomenute tehnologije takođe donose i nove izazove, kao što je, na primer, potreba za specijalizovanim metodologijama i test alatima koji su u stanju da odgovore na složenost IoT sistema i primena AI-ja i mašinskog učenja u softverskom testiranju (Hasan, 2022).

Veštačka inteligencija i mašinsko učenje u testiranju softvera: Integracija AI-ja i mašinskog učenja u softversko testiranje može da automatizuje kreiranje test slučajeva, unapredi prepoznavanje grešaka, predvidi kvalitet softvera i pruži dragocene uvide u sam proces testiranja (Džang i saradnici, 2020). Na primer, algoritmi mašinskog učenja mogu da analiziraju istorijske podatke kako bi predvideli oblasti podložne greškama i na taj način povećaju efikasnost procesa testiranja. Veštačka inteligencija, sa druge strane, može da automatizuje rutinske zadatke i smanji prostor za ljudske greške, čime se testerima omogućava da se fokusiraju na složenije zadatke.

Internet stvari (IoT) i softversko testiranje: Sve veća popularnost i dostupnost IoT uređaja povećala je složenost i opseg softverskog testiranja. Internet stvari predstavlja jedinstvenu kombinaciju hardvera i softvera, zbog čega zahteva testiranje u različitim uslovima i okruženjima kako bi se osiguralo njegovo nesmetano funkcionisanje. Da bi se to postiglo, neophodne su inovativne tehnike i test alati koji će biti u stanju da se nose sa povećanom složenošću i raznovrsnošću IoT sistema (Stol i saradnici, 2016).

Iako evolucija i razvoj ovih tehnologija donose nove izazove, one takođe stvaraju nove prilike za unapređenje efikasnosti i efektivnosti softverskog testiranja. Kontinuirano istraživanje i razvoj u ovoj oblasti su ključni ukoliko želimo da iskoristimo ove prilike i rešimo novonastale izazove.

Možemo da zaključimo da je testiranje softvera neophodna komponenta životnog ciklusa i razvoja softvera, budući da igra ključnu ulogu u osiguravanju kvaliteta i pouzdanosti softverskih proizvoda. Uprkos izazovima, kontinuirani razvoj i napredak metodologija testiranja i integracija novih tehnologija poput AI-ja, mašinskog učenja i Interneta stvari obećavaju efikasniju i uspešniju budućnost za softversko testiranje. Tekuća istraživanja i razvoj u ovoj oblasti su od ključnog značaja za prevazilaženje izazova i iskorišćavanje prilika, jer dovode do razvoja pouzdanijih i kvalitetnijih softverskih proizvoda. Sa razvojem tehnologije, razvijace se i naše test metode kako bi se osigurao kvalitet softvera, zbog čega nas u budućnosti očekuju kontinuirano učenje i adaptacija softverskog testiranja.

Zaključak

Testiranje softvera neodvojivi je deo razvoja softvera, budući da igra ključnu ulogu u osiguravanju kvaliteta i pouzdanosti softverskih proizvoda i značajno doprinosi zadovoljstvu i poverenju korisnika. Uprkos izazovima, kontinuirani razvoj metodologija i tehnologija za testiranje, naročito AI-ja, mašinskog učenja i Interneta stvari, grade temelj efikasnijeg i efektivnijeg softverskog testiranja. Kontinuirani razvoj i istraživanje u ovoj oblasti će nesumnjivo dovesti do razvoja pouzdanijih i kvalitetnijih softverskih proizvoda.

Zaključak koji se može izvući jeste da se oblast softverskog testiranja neprekidno razvija i prilagođava kako bi odgovorila na rastuću složenost softverskih aplikacija i rapidni tehnološki razvoj. Iako ove promene donose nove izazove, one takođe utiru put za napredak metodologija i tehnika testiranja koje će dovesti do razvoja pouzdanijih i kvalitetnijih softverskih proizvoda. Kontinuirano istraživanje i razvoj u ovoj oblasti su neophodni kako bismo odgovorili na te izazove i iskoristili potencijal novih tehnologija u testiranju softvera.

Literatura

1. Beizer, B. (1990). *Software testing techniques*. Van Nostrand Reinhold.
2. Divyani, G. (2020). A comprehensive approach to software testing. *International Journal of Advanced Research in Computer Science*, 9(2).
3. Fewster, M., & Graham, D. (1999). *Software test automation: effective use of test execution tools*. ACM Press/Addison-Wesley Publishing Co.
4. Hassan, S. (2022). Exploring the future of software testing: A comprehensive study. *Journal of Software: Evolution and Process*, 34(2).
5. Jorgensen, P. C. (2016). *Software testing: a craftsman's approach*. CRC press.
6. Kaner, C., Falk, J., & Nguyen, H. Q. (1993). *Testing computer software*. Wiley.
7. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.
8. Perry, W. E. (2006). *Effective methods for software testing*. John Wiley & Sons.
9. Pezze, M., & Young, M. (2007). *Software testing and analysis: process, principles, and techniques*. John Wiley & Sons.
10. Stol, K. J., Ralph, P., & Fitzgerald, B. (2016). Grounded theory in software engineering research: A critical review and guidelines. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)* (pp. 120–131). IEEE.
11. Zhang, Y., Li, Z., Wu, Y., Liang, B., & Yin, J. (2020). A Survey on the Application of Machine Learning in Software Testing. In *2020 35th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)* (pp. 14–19). IEEE.

