

Validacija upotrebljivosti ERP proizvoda

Krajinović, Magdalena

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:214737>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-25**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet Informatike u Puli

Magdalena Krajinović
VALIDACIJA UPOTREBLJIVOSTI ERP PROIZVODA

Diplomski rad

Pula, rujan 2024.

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike u Puli

Magdalena Krajinović
VALIDACIJA UPOTREBLJIVOSTI ERP PROIZVODA

Diplomski rad

JMBAG: 0303069253, izvanredna studentica

Studijski smjer: Informatika

Predmet: Napredne strukture i algoritmi

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv.prof.dr.sc. Tihomir Orehovački

Pula, rujan 2024.



IN2 društvo s ograničenom odgovornošću
za informatički inženjering i usluge

Marošnićeva 1/1
10000 Zagreb, HR

T +385 1 6386 800
F +385 1 6386 801

OIB 68195665956
MBS 080071082

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli
Rovinjska 14,
52100 Pula

Zagreb, 14.04.2023

PREDMET: Suglasnost za korištenje podataka u svrhu pisanja diplomskog rada studentice Magdalene Krajinović

Poštovani,

Temeljem zaprimljene zamolbe, koju nam je uputila studentica Magdalena Krajinović, o korištenju podataka i resursa poduzeća IN2 d.o.o. u svrhu pisanja i izrade diplomskog rada pod nazivom "Testiranje uporabe sučelja aplikacije „Korisnički portal“ koja je dio ERP sustava" (u daljnjem tekstu Diplomski rad), ovim putem potvrđujemo i dajemo našu suglasnost.

Suglasni smo da studentica Magdalena Krajinović koristi sve potrebne podatke i resurse našeg poduzeća u svrhu pisanja i izrade Diplomskog rada isključivo uz poštovanje sljedećih uvjeta:

1. Svi podaci i informacije koje joj IN2 pruži za potrebe Diplomskog rada tretiraju se kao vrlo povjerljivi. Studentica Magdalena Krajinović će osigurati i odgovorna je da se takvi podaci neće dijeliti ili otkrivati trećim stranama bez izričitog pisanog odobrenja poduzeća IN2.
2. Podaci tvrtke IN2 koje studentica Magdalena Krajinović koristi u Diplomskom radu bit će korišteni isključivo u svrhu izrade diplomskog rada na temu "Testiranje uporabe sučelja aplikacije „Korisnički portal“ koja je dio ERP sustava".
3. IN2 ni u kojem slučaju nije odgovoran i odriče se svake odgovornosti za bilo kakvu interpretaciju podataka koje je studentica Magdalena Krajinović napravila u sklopu Diplomskog rada. Studentica je isključivo odgovorna za sve aspekte rada, uključujući zakonitost i valjanost korištenih podataka.

Ova suglasnost se izdaje i vrijedi isključivo tijekom pisanja i izrade Diplomskog rada na temu "Testiranje uporabe sučelja aplikacije „Korisnički portal“ koja je dio ERP sustava" i danom obrane prestaje važiti, a svaka druga uporaba podataka poduzeća IN2 d.o.o. za bilo koji drugi projekt ili svrhu zahtijeva odvojeni pristanak poduzeća IN2 d.o.o. Zagreb.

Srdačan pozdrav,

Danijel Budaković

 **in2** informacijski
inženjering
IN2 d.o.o., Zagreb, Hrvatska

Izvršni direktor, Sektor zdravstva i javne uprave
Voditelj IN2 ureda Pula

IN2 d.o.o. • IN2 društvo s ograničenom odgovornošću za informatički inženjering i usluge • Marošnićeva 1/1., 10000 Zagreb, HR • T +385 1 6386 800 • F +385 1 6386 801 • Nadležni sud: Trgovački sud u Zagrebu • MBS 080071082 • OIB 68195665956 • PDV broj: HR68195665956 • Temeljni kapital: 2.030.658,97 € uplaćen u cijelosti • Uprava: Ivan Gabrić, Snježana Stanković, Sanja Svilokos • Banke/IBAN: Erste&Steiermarkische bank d.d., Rijeka/HR4124020061100054853 • Addiko Bank d.d., Zagreb/HR3525000091101256823 • Zagrebačka banka d.d., Zagreb/HR223600001101301489 • Privredna banka Zagreb d.d., Zagreb/HR7723400091110247069 • Raiffeisenbank Austria d.d., Zagreb/HR6824840081105429943 • www.in2.eu

Sadržaj

1. Uvod	1
2. ERP sustavi	3
3. Upotrebljivost	4
3.1 Upotrebljivost u ERP sustavima	4
4. Validacija upotrebljivosti	7
4.1 Vrednovanje kao dio validacije	8
4.2 Heuristička evaluacija	9
5. Kvaliteta softvera	12
6. Korisničko iskustvo i zadovoljstvo kupca	15
7. Životni ciklus testiranja	17
8. Norme i načela testiranja	22
9. Ručno testiranje	25
9.1 Prednosti ručnog testiranja u odnosu na automatsko	27
9.2 Uobičajene metrike ručnih testera	27
9.3 Vrste testiranja koje se izvode ručno	31
10. Validacija upotrebljivosti ERP proizvoda	34
10.1 Validacija upotrebljivosti web portala	35
10.2 Rezultati testiranja web portala	48
10.3 Validacija upotrebljivosti desktop i mobilne aplikacije	52
10.3.1 Rezultati testiranja desktop aplikacije	56
10.3.2 Rezultati testiranja mobilne aplikacije	59
Zaključak	62
Literatura	64
Sažetak	66
Abstract	67
Popis slika	68
Popis tablica	68

1. Uvod

U današnjem digitaliziranom svijetu, korisnici svakodnevno koriste razne vrste aplikacija i sustava. Od poslovnih softverskih rješenja do mobilnih aplikacija za svakodnevne zadatke. Upotrebljivost korisničkog sučelja ključna je za uspjeh ovih aplikacija jer određuje kako jednostavno, intuitivno i efikasno korisnici mogu izvršavati zadatke. Aplikacije koje su kompleksne za navigaciju ili su neprilagođene potrebama korisnika mogu dovesti do frustracije, smanjenja produktivnosti te gubitka samog korisnika. U kontekstu poslovnih rješenja poput ERP (eng. Enterprise Resource Planning) sustava, upotrebljivost postaje još važnija zbog složenosti sustava i širokog spektra korisnika s različitim razinama tehničke stručnosti. Stoga je ključan proces validacije upotrebljivosti ovih sustava kako bi se osiguralo da su dizajnirani prema stvarnim potrebama krajnjih korisnika, omogućujući im lakše obavljanje svakodnevnih zadataka bez nepotrebnih komplikacija. Tijekom procesa razvoja, utrošeni resursi za kontinuirano testiranje se često smatraju nepotrebnim. Često se smatraju i dodatnim troškom za poslovanje jer rezultati testiranja nisu toliko opipljivi kao razvoj softvera, stoga ih je teško opravdati. Kako bi se softver doveo do što bolje kvalitete, potrebno je provoditi testiranja jer ona pružaju načine mjerenja jesu li ispunjeni zahtjevi korisnika i ispunjava li kvaliteta softvera očekivanja.

Cilj ovog rada je validacija upotrebljivosti ERP sustava kroz provedbu testiranja sukladno ISO/IEC 25062:2006 (CIF) normi, s naglaskom na metode kao što je heuristička evaluacija koja omogućava rano otkrivanje problema u dizajnu korisničkog sučelja. Korištenje ove norme osigurava strukturirani pristup testiranju te omogućuje analizu učinkovitosti proizvoda iz perspektive krajnjih korisnika. Kroz validaciju upotrebljivosti je cilj dokazati ispunjava li proizvod zahtjeve korisnika i potvrditi njegovu učinkovitost nakon implementacije. Kroz analizu raznih pristupa i studija slučaja, prikazat će se kako validacija može značajno poboljšati korisničko iskustvo i povećati učinkovitost korištenja aplikacija.

Ovaj rad, osim Uvoda i Zaključka, sastoji se od osam poglavlja. U drugom poglavlju detaljnije se objašnjava pojam ERP sustava. Treće poglavlje prikazuje upotrebljivost te vrednovanje u testiranju ERP sustava. Četvrto poglavlje se bavi validacijom upotrebljivosti. Peto poglavlje obrađuje kvalitetu softvera. Šesto poglavlje bavi se

korisničkim iskustvom i zadovoljstvom kupaca. Sedmo poglavlje prikazuje životni ciklus testiranja. Osmo poglavlje objašnjava norme i načela testiranja. Deveto poglavlje bavi se ručnim testiranjem te prikazuje korištene metrike u empirijskom dijelu rada. Deseto poglavlje je empirijski dio rada, bavi se validacijom upotrebljivosti ERP proizvoda, prikazuje jednu studiju slučaja testiranja web proizvoda te dvije studije slučaja testiranja aplikacije, u mobilnoj i desktop verziji.

2. ERP sustavi

ERP sustavi su softverski sustavi koji integriraju i optimiziraju poslovne procese unutar poduzeća. Omogućuju trenutan pristup informacijama koji poboljšava suradnju i komunikaciju unutar cijelog poduzeća (Bradford, 2015). Koriste se u velikim poduzećima za podršku većini ili svim njihovim funkcijama (Sommerville, 2016). Organizacije nastoje optimizirati svoje poslovne procese, a jedan od najboljih načina za postizanje tog cilja je implementacija sustava za planiranje resursa poduzeća. ERP sustavi su se razvili posljednjih desetljeća kako bi poboljšali poslovne procese organizacija i povećali njihovu učinkovitost. Ovi sustavi imaju različite i fleksibilne karakteristike i mogu pružiti prilike za rast i održivost velikim tvrtkama, ali i malim i srednjim poduzećima. ERP sustav je u središtu institucije. On omogućuje integraciju i optimizaciju poslovnih procesa unutar organizacije putem jedinstvene baze podataka koja podržava sve ključne funkcije, uključujući nabavu, računovodstvo i prodaju (Bradford, 2015). Bavi se ključnim zadacima upravljanja i integracije poslovnih procesa u stvarnom vremenu. Značajke koje treba uzeti u obzir pri odabiru najprikladnijeg ERP rješenja su fleksibilnost, modularnost, povezanost, inovativna tehnologija, Cloud i SaaS okruženje te jednostavnost korištenja. ERP sustav je povezan s korisničkim sučeljem i dizajniran je za pružanje mogućnosti obrade informacija za podršku u strategiji, operacijama, analizi upravljanja i funkcijama donošenja odluka u organizaciji. Korisnik je u središtu informacijskog sustava (Matende i Ogao, 2013).

Visoka kvaliteta ERP sustava neophodna je kako bi se osigurala stabilnost, pouzdanost i sigurnost poslovnih procesa unutar organizacije. Pruža integrirani pristup upravljanju resursima, čime se smanjuje operativna složenost i poboljšava učinkovitost. ERP sustavi igraju ključnu ulogu u smanjenju dupliciranja podataka i unaprjeđenju transparentnosti poslovnih operacija, što omogućuje bolju kontrolu i predviđanje poslovnih rezultata (Sommerville, 2016).

3. Upotrebljivost

Upotrebljivost je novo svojstvo koje ovisi o interakcijama među korisnicima, proizvodima, zadacima i okruženjima (Lewis i Sauro, 2021). Ona je mjera u kojoj određeni korisnici mogu koristiti proizvod za postizanje određenih ciljeva efektivno, učinkovito i sa zadovoljstvom u određenom kontekstu korištenja.

Upotrebljivost u informatici se odnosi na lakoću korištenja određenog softverskog sustava, aplikacije ili web stranice. Također se odnosi na lakoću korištenja softverskog sustava. Dobar dizajn povećava učinkovitost korisnika, dok loše dizajnirani sustavi smanjuju produktivnost i izazivaju frustracije (Nielsen, 1994). Upotrebljivost također uključuje sposobnost sustava da zadovolji potrebe korisnika u kontekstu njegove primjene (Garrett, 2010). Primarni cilj dizajnera i programera u procesu razvoja softvera je osigurati korisnicima da mogu lako i intuitivno ostvariti svoje ciljeve, dok će tester bilježiti zadovoljavanje li softver korisničke zahtjeve i njegovu upotrebljivost. Upotrebljivost uključuje različite aspekte, poput jednostavnosti navigacije, brzine obavljanja zadataka, razumljivosti povratnih informacija, te sveukupnog korisničkog iskustva. Jedan od najpoznatijih okvira za procjenu upotrebljivosti dolazi od Jakoba Nielsena, koji je razvio 10 heurističkih pravila za upotrebljivost. Ova pravila uključuju aspekte poput povratnih informacija sustava, fleksibilnosti i efikasnosti korištenja, te prevencije grešaka, što su ključni čimbenici u stvaranju funkcionalnog softverskog sučelja.

3.1 Upotrebljivost u ERP sustavima

Upotrebljivost se odnosi na jednostavnost i prirodost dijaloga između korisnika i sustava, čime se minimiziraju korisničke greške i povećava zadovoljstvo (Nielsen, 1994). Kada govorimo o ERP sustavima, upotrebljivost postaje još važnija zbog složenosti sustava. ERP sustavi objedinjuju različite poslovne funkcije poput financija, nabave, ljudskih resursa i proizvodnje, što znači da se različite vrste korisnika, s različitim stupnjevima tehničke stručnosti, oslanjaju na isti sustav.

Ključni aspekti upotrebljivosti ERP sustava uključuju:

- Jednostavnost korištenja jer ERP sustavi često imaju više modula i kompleksne funkcionalnosti, pa je važno da korisnici mogu brzo i efikasno obavljati zadatke.
- Prilagođenost korisničkom profilu jer ERP sustavi moraju biti dovoljno fleksibilni kako bi zadovoljili potrebe različitih tipova korisnika (od stručnjaka do administrativnog osoblja).
- Efikasnost, uzimajući u obzir obujam zadataka koje ERP sustavi obuhvaćaju, korisnicima mora biti omogućeno da u što kraćem roku završe potrebne procese, bez suvišnih koraka i tehničkih problema.

Učinkovito testiranje upotrebljivosti ERP sustava često uključuje heurističku evaluaciju i testiranje korisnika, kako bi se uočile nepravilnosti u dizajnu sustava i kako bi se osiguralo optimalno korisničko iskustvo. Istraživanje koje je provelo Sveučilište u Valenciji 2013. godine analiziralo je upotrebljivost ERP sustava u malim i srednjim poduzećima. U istraživanju je testirano 20 različitih ERP sustava s naglaskom na korisničko iskustvo. Rezultati su pokazali da su ključni izazovi bili prekomplikirana sučelja i potreba za dugotrajnim obukama zaposlenika kako bi se sustavi ispravno koristili. Istraživači su zaključili da su ERP sustavi previše orijentirani na funkcionalnosti, a premalo na intuitivnost što otežava njihovo korištenje u svakodnevnim poslovnim operacijama (Hvannberg et al., 2007). Navedeno istraživanje naglašava važnost validacije upotrebljivosti u ERP sustavima, jer složeni poslovni procesi ne smiju ugrožavati korisničko iskustvo, već trebaju omogućiti lakše i efikasnije obavljanje zadataka.

U nastavku slijedi nekoliko primjera ručnog testiranja validacije upotrebljivosti koji se primjenjuju u različitim industrijama i softverskim proizvodima:

1. Testiranje upotrebljivosti ERP sustava u financijama

U testiranju ERP sustava koji se koristi za financijske operacije, ručno testiranje validacije upotrebljivosti može uključivati procese poput unosa faktura, obračuna plaća, upravljanja troškovima te kreiranja financijskih izvještaja. Tester ručno prolaze kroz korake potrebne za svaku od tih funkcionalnosti kako bi provjerili koliko je sustav intuitivan, koliko su jednostavni meniji i koliko brzo korisnik može obaviti zadatak bez pogrešaka ili zbunjenosti.

2. Testiranje upotrebljivosti CRM modula u ERP sustavu

Ovdje se testerima usredotočuju na korisničko iskustvo u upravljanju odnosima s klijentima. Testiranje validacije upotrebljivosti može uključivati unos novih kontakata, praćenje prodajnih prilika ili upravljanje komunikacijom s klijentima. Testerima procjenjuju koliko je sučelje jednostavno za navigaciju te koliko su funkcije logično organizirane za krajnjeg korisnika.

3. Testiranje e-trgovine unutar ERP sustava

Ako ERP sustav ima modul za e-trgovinu, ručno testiranje upotrebljivosti može uključivati simulaciju procesa online naručivanja, provjeru stanja zaliha, kreiranje i potvrđivanje narudžbi, te upravljanje dostavom. Testerima provjeravaju kako korisnici mogu upravljati procesima te koliko su razumljive povratne informacije sustava (npr. poruke o uspješno obrađenoj narudžbi).

4. Testiranje upotrebljivosti u industrijskim ERP sustavima

U industrijskim ili proizvodnim ERP sustavima, ručno testiranje validacije upotrebljivosti uključuje procjenu modula za upravljanje proizvodnjom, planiranje resursa i praćenje proizvodnih procesa. Testerima procjenjuju koliko je sustav prilagođen operaterima u stvarnim uvjetima, poput upravljanja strojevima ili optimizacije radnih procesa.

5. Testiranje u ERP sustavima za ljudske resurse (HR)

Ovdje testerima ručno prolaze kroz procese unosa podataka o zaposlenicima, obračun plaća, praćenje odsutnosti ili evaluaciju radnog učinka. Fokus je na tome koliko je jednostavno za HR menadžere i zaposlenike da koriste sustav za svakodnevne zadatke bez dodatne tehničke podrške.

Ključne točke validacije upotrebljivosti u ovim primjerima su jednostavnost navigacije u kojoj se provjerava koliko je sučelje intuitivno i lako za korištenje. U validiranju efikasnosti obavljanja zadataka mjeri se koliko korisnici mogu brzo i točno završiti zadatak. ERP sustavi su prilagođeni različitim korisničkim profilima jer ih često koriste različiti zaposlenici s različitim stupnjevima tehničke stručnosti, stoga je ključno validirati upotrebljivost za sve tipove korisnika. Validacija upotrebljivosti smanjuje greške jer provjerava koliko je sustav otporan na korisničke greške i koliko brzo sustav daje povratne informacije. Ovi primjeri pokazuju kako se ručno testiranje validacije upotrebljivosti koristi u različitim modulima ERP sustava kako bi se osigurala što bolja prilagodba stvarnim potrebama korisnika.

4. Validacija upotrebljivosti

Validacija upotrebljivosti predstavlja proces u kojem se provjerava koliko dobro sustav zadovoljava zahtjeve korisnika u stvarnim uvjetima korištenja. Ovaj proces uključuje prikupljanje povratnih informacija od korisnika, identifikaciju problema kroz testiranje te prilagodbu aplikacije na temelju prikupljenih podataka. Cilj validacije osigurava tehničku ispravnost i potvrđuje da aplikacija ispunjava poslovne ciljeve, te da je prilagođena stvarnim potrebama korisnika. Testiranje softvera neophodan je procesu razvoja poslovnih aplikacija. Pravilno testiranje omogućuje pravovremeno otkrivanje grešaka i poboljšanje upotrebljivosti proizvoda, čime osigurava njegovo ispravno funkcioniranje prije konačne implementacije.

Validacija softvera je proces kojim se osigurava da softver ispunjava specifikacije. Ona mora osigurati da sustav odgovara stvarnim potrebama korisnika te da proizvod zadovoljava stvarne potrebe korisnika, a ne samo tehničke specifikacije (Sommerville, 2016).

Validacija upotrebljivosti često uključuje metode evaluacije upotrebljivosti kao što je ručno testiranje koje je ključno u ERP sustavima kako bi se osigurala funkcionalnost i jednostavnost korisnika. Proces evaluacije upotrebljivosti obuhvaća različite tehnike, uključujući heurističku procjenu i testiranje usmjereno na korisnika, s ciljem mjerenja djelotvornosti, učinkovitosti i zadovoljstva korisnika u interakciji sa sustavom. Vrednovanje se koristi u validaciji upotrebljivosti i ima ključnu ulogu u procjeni kvalitete korisničkog iskustva. Kada govorimo o validaciji upotrebljivosti, vrednovanje se odnosi na proces prikupljanja i analize podataka o tome kakvu interakciju korisnici imaju sa softverskim sustavom i koliko dobro sustav zadovoljava njihove potrebe.

U ERP sustavima upotrebljivost je važna zbog složenosti i kritične prirode softverskih rješenja u poslovnim procesima. Sigurnost da je ERP sustav jednostavan za korištenje može uvelike utjecati na stope usvajanja, produktivnost korisnika i uspjeh sustava. Istraživanja su pokazala da testiranje upotrebljivosti, uključujući metode poput heurističkih procjena, igra značajnu ulogu u poboljšanju korisničkih sučelja i tijekom rada unutar ERP sustava (Bastardo et al., 2024).

Načini na koje možemo provoditi vrednovanja u validaciji upotrebljivosti su:

1. Heuristička evaluacija

Ovu metodu koriste stručnjaci za pregledavanje korisničkog sučelja prema unaprijed definiranim pravilima upotrebljivosti (npr. Nielsenovih 10 heuristika). Cilj im je identificirati probleme koji smanjuju kvalitetu korisničkog iskustva (Nielsen, 1994a).

2. Korisničko testiranje

Tijekom ovog procesa stvarni korisnici prolaze kroz ključne zadatke unutar sustava. Njihove performanse i zadovoljstvo se vrednuju kako bi se utvrdilo koliko je sustav upotrebljiv.

3. Ankete i upitnici

Ankete i upitnici se koriste za prikupljanje povratnih informacija o percepciji korisnika o sustavu, poput zadovoljstva, jednostavnosti korištenja ili frustracija koje su iskusili tijekom interakcije.

4. Analiza ponašanja

Analizom ponašanja se proučava ponašanje korisnika kroz metrike kao što su brzina dovršavanja zadataka, broj pogrešaka i razina podrške potrebne korisnicima, što su ključni aspekti u vrednovanju upotrebljivosti.

4.1 Vrednovanje kao dio validacije

Validacija upotrebljivosti uključuje potvrđivanje da sustav zadovoljava potrebe korisnika, a vrednovanje pomaže u kvantificiranju tih aspekata i identificiranju potencijalnih problema. Vrednovanje omogućuje dizajnerima i programerima da mjere efikasnost, djelotvornost i zadovoljstvo korisnika, čime se osigurava da softver ispunjava ciljeve upotrebljivosti i poslovne ciljeve te povećava kvalitetu proizvoda. Vrednovanje je integralni dio validacije upotrebljivosti jer pomaže identificirati i procijeniti probleme s korisničkim iskustvom te osigurava prilagodbu sustava potrebama korisnika.

Kako bi se smanjili potencijalni nedostaci različitih metoda upotrebljivosti, preporučuje se njihova kombinacija prilikom vrednovanja, kako unutar pojedinih kategorija, tako i među njima. Pri odabiru metode važno je uzeti u obzir različite kriterije, uključujući potrebne resurse (poput vremena, proračuna, broja evaluatora i njihove stručnosti,

broja korisnika te opreme za testiranje), razinu objektivnosti, kao i mogućnost primjene u različitim fazama razvoja web aplikacije (Plantak Vukovac i Orehovački, 2010).

4.2 Heuristička evaluacija

Heuristička procjena je proces u kojem stručnjaci koriste osnovna pravila za mjerenje upotrebljivosti korisničkih sučelja dok testiraju proizvod i evidentiraju potencijalne probleme i greške. Evaluatori koriste utvrđenu heuristiku (npr. Nielsen-Molichovu) i otkrivaju uvide koji mogu pomoći razvojnom timu da poboljša upotrebljivost proizvoda još u ranom razvoju (Interaction Design Foundation, 2016a).

Da biste proveli heurističku procjenu trebate znati što testirati i kako testirati. Bilo da se radi o cijelom proizvodu ili jednom postupku, treba jasno definirati parametre onoga što se testira i što je cilj. Bitno je upoznati svoje korisnike i imati na umu jasne definicije ciljeva ciljne publike, konteksta itd. Osobine korisnika pomažu evaluatorima testirati proizvod iz perspektive korisnika. Najbolje je odabrati 3 do 5 stručnih evaluatora u upotrebljivosti i relevantnom polju. Ukratko im objasniti koje dijelove trebaju pokriti u odabiru zadataka Ovisno o prirodi proizvoda je potrebno definirati heuristiku (Nielsen-Molich heuristike i/ili definiranje drugih). U sesiji se predlaže ispitati evaluatore kako bi se mogli usporediti rezultati za analizu i kako bi se predložili popravci.

Primjer heurističke procjene je kada stručnjaci za upotrebljivost procjenjuju web mjesto ili aplikaciju u odnosu na utvrđena načela upotrebljivosti ili heuristike, kako bi identificirali potencijalne probleme s korisničkim iskustvom.

Pri korištenju heurističke evaluacije treba biti oprezan. Ako u testiranju sudjeluje jedan evaluator, on može propustiti grešku koja mu nije odmah očita. Stoga se preporuča da u testiranju sudjeluje više evaluatora. Budući da su iskusni evaluatori skuplji, teško se dolazi do njih i zato se ponekad moraju uzeti manje stručni ljudi. To znači da bi i rezultati testiranja mogli biti manje vrijedni, ali to nije obeshrabrujuće jer je heuristička analiza uvijek otvorena za raspravu (Interaction Design Foundation, 2016b).

Prednosti heurističke evaluacije mogu pomoći u isticanju potencijalnih problema upotrebljivosti rano u procesu dizajna. Brz je i jeftin alat u usporedbi s drugim metodama koje uključuju stvarne korisnike. Ali, heuristička evaluacija ovisi o znanju i stručnosti evaluatora jer edukacija evaluatora ili angažiranje vanjskih povećava vrijeme i novac potreban za provođenje evaluacije. Oni s manjim budžetima si ne mogu priuštiti

takve poteze. Heuristička procjena se temelji na pretpostavkama o tome što je 'dobra' upotrebljivost, a kako se heuristika temelji na istraživanju, to je često točno. Međutim, evaluacije nisu zamjena za testiranje sa stvarnim korisnicima. Ovo su samo smjernice, a ne rigidna pravila. Heuristička procjena također može završiti davanjem lažnih uzbuna. Prema Wongu je otkriveno da 43% problema identificiranih eksperimentalnim heurističkim procjenama zapravo nisu problemi. Nadalje, evaluatori su mogli identificirati samo 21% stvarnih problema upotrebljivosti u usporedbi s testiranjem upotrebljivosti (Wong, 2024).

U testiranju ERP sustava je najbolje koristiti heurističke evaluacije koje su prilagođene složenosti i specifičnostima takvih softverskih rješenja. Neke od najprikladnijih heuristika za ERP sustave uključuju vidljivost statusa sustava, podudaranje između sustava i stvarnog svijeta, norme i konzistentnost, prevencija grešaka te podrška fleksibilnosti i efikasnosti. U nastavku rada su pojmovi navedenih heurističkih evaluacija detaljnije navedeni.

Vidljivost statusa sustava

Korisnici trebaju biti jasno obaviješteni o statusu sustava u svakom trenutku. ERP sustavi obuhvaćaju složene poslovne procese, pa je važno da korisnici imaju uvid u trenutne operacije i njihov napredak (Nielsen, 1994).

Podudaranje između sustava i stvarnog svijeta

ERP sustavi umjesto tehničkih pojmova trebaju koristiti jezik i termine koji su razumljivi korisnicima i koji odražavaju stvarne poslovne procese (Nielsen, 1994).

Konzistentnost i norme

Budući da različiti dijelovi sustava često pokrivaju različite funkcionalnosti i poslovne procese, ERP sustavi moraju biti dosljedni u svom dizajnu i primjeni pravila kako bi se izbjegla zabuna korisnika (Kaner & Bach, 2001).

Prevencija grešaka

Ključ za održavanje integriteta podataka je kada ERP sustav omogući mehanizme koji korisnicima pomažu izbjegavanje grešaka, posebno u unosu podataka ili postavljanju složenih poslovnih operacija (Nielsen, 1994).

Podrška fleksibilnosti i efikasnosti

ERP sustavi trebaju biti dovoljno jednostavni za povremene korisnike, ali također trebaju omogućiti naprednim korisnicima prilagodbu sučelja ili postavki kako bi povećali svoju učinkovitost (Nielsen, 1994).

Heuristička evaluacija temeljena na ovim principima pomaže identificirati glavne probleme upotrebljivosti unutar ERP sustava te omogućuje prilagodbu softvera krajnjim korisnicima kako bi se poboljšalo ukupno korisničko iskustvo.

5. Kvaliteta softvera

Kvaliteta softvera obuhvaća mnoge aspekte, uključujući pouzdanost i održivost, a održavanje visoke kvalitete softvera kroz kontinuirano testiranje je ključno za uspješnu implementaciju (Kaner i Bach, 2001). Kvaliteta ima dvije radne definicije. Prva kaže da je gledište na proizvod proizvođača da isti ispunjava zahtjeve, a druga kaže da je gledište na proizvod kupca da isti zadovoljava njegove potrebe i prikladan mu je za uporabu (PeopleCert International Ltd 2023).

Definicija 'kvalitete' je čimbenik u određivanju opsega testiranja softvera. Iako postoji više definicija kvalitete, važno je napomenuti da većina sadrži iste ključne komponente:

- Kvaliteta se temelji na zadovoljstvu kupaca.
- Vaša organizacija mora definirati kvalitetu prije nego što se ona može postići.
- Uprava mora voditi organizaciju kroz sve napore za poboljšanje kvalitete.

Kvaliteta softvera uključuje atribute poput ispravnosti, pouzdanosti i održivosti (Pressman, 2015).

Postoji pet perspektiva kvalitete, od kojih se svaka treba smatrati važnom za kupca:

1. Transcendentno; znam kad to vidim
2. Temeljeno na proizvodu; posjeduje željene značajke
3. Prilagođenost korisniku za korištenje
4. Razvoj i proizvodnja; u skladu sa zahtjevima
5. Na temelju vrijednosti po prihvatljivoj cijeni

Kvaliteta softvera uključuje i funkcionalnost i druge nefunkcionalne osobine, poput pouzdanosti i održivosti, koji značajno utječu na korisničko iskustvo (Sommerville, 2016). Patrick Townsend ispituje kvalitetu u stvari i kvalitetu u percepciji kao što je prikazano u tablici 1.1. Kvaliteta je zapravo obično gledište dobavljača, dok je kvaliteta u percepciji kupca. Svaka razlika između prvog i drugog može uzrokovati probleme između njih dvoje.

Tablica 1 Townsend-ov pogled na kvalitetu, izvor: izrada autora prema PeopleCert International Ltd 2023, str. 12

KVALITETA U STVARNOSTI	KVALITETA U PERCEPCIJI
Raditi ispravnu stvar.	Isporučiti pravi proizvod.
Raditi na ispravan način.	Zadovoljiti potrebe naših kupaca.
Napraviti ispravno iz prvog puta.	Ispuniti očekivanja kupaca.
Napraviti na vrijeme.	Ponašati se prema svakom kupcu s integritetom, ljubaznošću i poštovanjem.

Politika kvalitete organizacije mora definirati i sagledati kvalitetu iz perspektive kupaca. Ako postoje sukobi, oni se moraju riješiti kako bi proizvod bio u skladu s očekivanjima. Kao što je ranije spomenuto, postoje dvije važne definicije kvalitetnog softvera:

- Proizvođačev pogled na kvalitetan softver znači ispunjavanje zahtjeva.
- Korisnikov pogled na kvalitetan softver znači prikladan za upotrebu.

Ove dvije definicije nisu nedosljedne. Ispunjavanje zahtjeva proizvođačeva je definicija kvalitete; to znači da proizvođač razvija softver u skladu sa zahtjevima. Definicija prikladnosti za upotrebu korisnikova je definicija kvalitete softvera; to znači da softver koji je razvio proizvođač zadovoljava potrebe korisnika bez obzira na zahtjeve softvera. U većini IT grupa postoje dvije praznine kao što je prikazano na slici 1. Ove praznine predstavljaju različite perspektive kvalitete softvera kako ih vide proizvođač i kupac.



Slika 1 Razlika u kvaliteti softvera između proizvođača i korisnika, izvor: PeopleCert International Ltd 2023, str. 13

Prvi jaz je jaz proizvođača. To je jaz između onoga što je određeno za isporuku, što znači dokumentirane zahtjeve i interne IT norme, i onoga što je stvarno isporučeno. Drugi jaz je između onoga što je proizvođač stvarno isporučio i onoga što je kupac očekivao.

Značajna uloga testiranja softvera pomaže u uklanjanju ta dva jaza. IT funkcija kvalitete prvo mora poboljšati procese do točke u kojoj IT može proizvoditi softver prema primljenim zahtjevima i svojim internim normama. Cilj funkcije kvalitete kojom se zatvara jaz između proizvođača jest omogućiti IT funkciji da osigura dosljednost u onome što može proizvesti. To se naziva "McDonald's efekt". To znači da kada uđete u bilo koji McDonald's na svijetu, Big Mac bi trebao imati isti okus. To ne znači da se vama kao kupcu sviđa Big Mac ili da on zadovoljava vaše potrebe, već da je McDonald's sada postigao dosljednost u isporučenom proizvodu.

Kako bi zatvorio prazninu korisnika, funkcija IT kvalitete mora razumjeti stvarne potrebe korisnika. To se može učiniti na sljedeći način:

- ankete kupaca
- zajednički razvoj aplikacija - sesije na kojima se proizvođač i korisnik sastaju te pregovaraju i slažu se oko zahtjeva
- veća uključenost korisnika tijekom izgradnje informacijskih proizvoda
- provođenje agilnih razvojnih strategija.

Kvaliteta softvera može se mjeriti kroz kombinaciju funkcionalnosti, učinkovitosti i pouzdanosti sustava, što zajedno čini uspješan softverski proizvod (Pressman, 2020). Kvaliteta softvera u ERP sustavu ključna je za njegovo učinkovito funkcioniranje i uspješnu integraciju u poslovne procese. ERP sustavi obuhvaćaju različite poslovne funkcije, te svaka greška ili nedostatak u kvaliteti može uzrokovati značajne operative probleme, uključujući smanjenje produktivnosti, gubitak podataka ili narušeno korisničko iskustvo. Kvalitetan ERP sustav omogućuje bolju koordinaciju i automatizaciju poslovnih procesa te doprinosi povećanju organizacijske efikasnosti kroz bolju dostupnost i upravljanje podacima (Bradford, 2015). Visoka kvaliteta softvera osigurava stabilnost, pouzdanost i sigurnost sustava, što omogućuje nesmetanu integraciju podataka i olakšava donošenje strateških odluka.

6. Korisničko iskustvo i zadovoljstvo kupca

Korisničko iskustvo (eng. User experience) je ključan aspekt razvoja softvera. Uspješan dizajn korisničkog iskustva mora biti usmjeren na korisnikove potrebe i ciljeve (Garrett, 2010). Zadovoljstvo korisnika ovisi o tome kako je softver prilagođen stvarnim potrebama korisnika (Nielsen, 1994).

Istraživanja o korisničkom iskustvu koja nadilaze klasične aspekte upotrebljivosti imaju povijest dugu otprilike dva desetljeća. Tradicionalna upotrebljivost se usredotočuje na instrumentalne značajke poput učinkovitosti i djelotvornosti, dok korisničko iskustvo donosi širi okvir, obuhvaćajući emocionalne i psihološke elemente doživljaja korisnika. Korisničko iskustvo nije samo funkcionalnost sustava, već uključuje i emocionalne reakcije, poput povjerenja i zadovoljstva, koje direktno utječu na uspjeh proizvoda (Garrett, 2010). Učinkovitost i djelotvornost i dalje su važni, no prvenstveno zato što imaju direktan utjecaj na emocionalne aspekte kao što su zadovoljstvo, povjerenje te percipirana estetska privlačnost s proizvodom. Ove emocionalne reakcije značajno oblikuju korisničko ponašanje, poput odluka o ponovnom korištenju proizvoda ili njegovoj preporuci drugima. U suštini, korisničko iskustvo istražuje šire aspekte doživljaja, uključujući emocionalne i estetske komponente, koje često presudno utječu na uspjeh proizvoda.

Zadovoljstvo kupca igra ključnu ulogu u uspjehu proizvoda u informatici, gdje se očekivanja korisnika često brzo mijenjaju s obzirom na stalne tehnološke inovacije. Zadovoljstvo kupca definira se kao percepcija korisnika o tome koliko proizvod ispunjava ili premašuje njihova očekivanja. U kontekstu informatičkih proizvoda, zadovoljstvo kupca može se mjeriti kroz različite parametre, uključujući jednostavnost korištenja, funkcionalnost, pouzdanost, brzinu izvršavanja zadataka te podršku korisnicima. Podrška korisnicima je način na koji tim korisničke službe osigurava da njihovi postupci proizvode dosljedne rezultate zadovoljnih kupaca. Razine zadovoljstva kupaca obično se utvrđuju postavljanjem pitanja telefonom ili naknadnim slanjem ankete. U nastavku su prikazane neke od prednosti pozitivnog zadovoljstva kupaca.

1. Smanjuje odljev kupaca
2. Povećava lojalnost kupaca i stalni prihod

3. Poboljšava učinkovitost i produktivnost
4. Povećava zadovoljstvo zaposlenika
5. Potiče zagovaranje marke
6. Pomaže vam da se izdvojite od konkurencije
7. Pruža mogućnosti povećanja prodaje i dodatne prodaje
8. Koristi drugim odjelima

Nemoguće je stvoriti okruženje u kojem su kupci i agenti zadovoljni 100% vremena. Ali potrebno je uzeti u obzir postavljanje i postizanje ciljeva za ovu ključnu metriku korisničke usluge, istovremeno osiguravajući da se klijenti i agenti osjećaju cijenjenima i shvaćenima (Kaizo, 2023).

Ključni faktori koji doprinose zadovoljstvu kupca informatičkog proizvoda uključuju intuitivno sučelje, brzinu i učinkovitost softvera, kompatibilnost s drugim sustavima i aplikacijama, te dostupnost kvalitetne korisničke podrške. Dodatno, redovita ažuriranja i nadogradnje proizvoda, koja odgovaraju na povratne informacije korisnika i ispravljaju eventualne nedostatke, također doprinose dugoročnom zadovoljstvu kupaca. Zadovoljstvo kupaca može se poboljšati ako dobro razumijemo svoju ciljanu publiku i njihove potrebe kako bismo mogli prilagoditi svoju ponudu proizvoda u skladu s tim. To će nam pomoći da stvorimo proizvod koji je uistinu jedinstven i cijenjen od strane naših kupaca. Također će nam pomoći shvatiti zašto su naši kupci zadovoljni ili nezadovoljni našom ponudom proizvoda. Zadovoljstvo kupca u području informatike nije samo mjera trenutnog uspjeha proizvoda, već i ključan pokazatelj njegove buduće održivosti na tržištu. Razumijevanje i upravljanje korisničkim zadovoljstvom dovodi do poboljšanja korisničkog iskustva te rezultira dugoročnim poslovnim uspjesima organizacije i jačanju lojalnosti kupaca.

7. Životni ciklus testiranja

Životni ciklus testiranja softvera je temeljni dio životnog ciklusa razvoja softvera (eng. software development life cycle) koji se sastoji samo od faze testiranja i počinje čim korisnički zahtjevi budu definirani. Životni ciklus testiranja softvera obuhvaća različite faze od analize zahtjeva do izvršenja testova. Testiranje nije jednokratan proces; ono traje tijekom cijelog životnog ciklusa softverskog proizvoda (Pressman, 2015). Ono mora biti planirano kroz sve faze životnog ciklusa kako bi se osigurala što veća pokrivenost i smanjenje pogrešaka (Pressman, 2020). Glavni cilj je identificirati i dokumentirati sve nedostatke ili probleme u softverskom proizvodu što je ranije moguće. U početnim fazama životnog ciklusa dok je proizvod još u razvoju, tim za testiranje analizira i definira opseg testiranja, ulazne i izlazne kriterije i testne slučajeve. Navedeno pomaže smanjiti vrijeme ispitnog ciklusa i poboljšati kvalitetu proizvoda. Životni ciklus testiranja uključuje kontinuirano testiranje sustava tijekom cijelog procesa razvoja. Testiranje punog životnog vijeka uključuje testove provjere i testove validacije. Životni ciklus testiranja ne može se provesti dok se ne usvoji formalizirani pristup životnog ciklusa. Isti ovisi o završetku unaprijed određenih isporuka u određenim točkama životnog ciklusa razvoja. Ako postoji značajna varijabilnost u razvojnim procesima, vrlo je teško učinkovito testirati i izvršne i neizvršne isporučene rezultate. Faze životnog ciklusa testiranja softvera uključuju analiziranje zahtjeva nakon kojeg slijedi izrada testnog planiranja. Zatim slijedi kreiranje testnih scenarija i slučajeva te na posljetku njihovo izvođenje. Dobiveni rezultati se dokumentiraju i evidentirane greške se prijavljuju. Nakon ispravka greške vrši se ponovna provjera testa.

Svaka faza testiranja mora biti planirana kako bi se osigurala što veća pokrivenost testovima (Kaner i Bach, 2001). Postoji šest glavnih faza u svakom modelu životnog ciklusa testiranja softvera (STLC model): analiza zahtjeva, planiranje testiranja, razvoj testnog slučaja, postavljanje testnog okruženja, izvršenje testa i zatvaranje ciklusa ispitivanja. Svaka od ovih faza ima određene ulazne i izlazne kriterije, aktivnosti i rezultate povezane s njom. Ulazni kriteriji daju preduvjetne stavke koje se moraju ispuniti prije početka testiranja, a izlazni kriteriji definiraju stavke koje se moraju ispuniti prije nego što se testiranje može zaključiti (Hamilton, 2024). U nastavku je detaljnije opisano svih šest faza životnog ciklusa testiranja softvera.

1. Analiza zahtjeva

Faza testiranja zahtjeva poznata i kao analiza zahtjeva u kojoj ispitni tim proučava zahtjeve sa stajališta testiranja kako bi identificirao zahtjeve koji se mogu testirati, a tim inženjera osiguranja kvalitete može komunicirati s različitim dionicima kako bi detaljno razumio zahtjeve. Zahtjevi mogu biti funkcionalni ili nefunkcionalni. Izvedivost automatizacije za projekt testiranja također se radi u ovoj fazi.

Specifikacija softverskih zahtjeva (SRS) je dokument koji sadrži potpuni opis očekivanih performansi sustava. Pregled SRS-a je prolaženje kroz dokument specifikacije funkcionalnih zahtjeva i pokušaj razumijevanja kakav će biti ciljani proizvod. Izazov dokumenta specifikacije je da nikada ne možemo prikupiti sve informacije odjednom. Potrebno je održati mnogo sastanaka i rasprava kako bi se dokument specifikacije usavršio. Tijekom rasprava je obavezno raditi bilješke radi daljnjeg praćenja, a i kako ne bi bilo propusta. Neki proizvodi se dugo razvijaju i skloni su izmjenama zahtjeva zato je potrebno imati prijašnju verziju dokumenta za usporedbu. Također, vrijeme za rasprave treba planski definirati jer bi ponekad brzina rasprave bila prebrza ili prespora za razumijevanje.

Aktivnosti u fazi testiranja zahtjeva:

- Identificiranje vrste testova koje treba provesti
- Prikupljanje detalja o prioritetima testiranja i fokusu
- Pripremanje matrice sljedivosti zahtjeva (RTM)
- Identificiranje pojedinosti testnog okruženja u kojem se testiranje treba provesti
- Analiziranje izvedivosti automatizacije (ako je potrebna)

2. Planiranje testiranja

Testiranje mora biti plansko i organizirano. U većini projekata, a naročito u koje je uključeno više ljudi, voditelj testiranja izrađuje dokument plana testiranja. Ovaj dokument se kreira kako bi svi menadžeri, tester i programeri bili informirani o planu i testnim scenarijima. Plan testiranja sadrži detaljno razumijevanje tijekom rada aplikacije. Sastoji se od testnih predložaka koji imaju uvod, opseg, strategiju testiranja, zahtjeve okoline, raspored testiranja, funkcije koje se testiraju, resurse i odgovornost, korake za reprodukciju i koji je rezultat izlaza, kriterije nastavka, ovisnosti, rizike, alate,

dokumentaciju i odobrenja. Ova faza je važna u testiranju jer nam o njoj ovisi rezultat uz koji će se pokazati i razumijevanje ili nerazumijevanje analize zahtjeva. Najbolji način testa je stvoriti matično okruženje gdje se softver može testirati u svim okruženjima.

Aktivnosti planiranja testiranja:

- Priprema testnog plana/strateškog dokumenta za različite vrste testiranja
- Izbor testnog alata
- Procjena napora testa
- Planiranje resursa i određivanje uloga i odgovornosti.
- Zahtjev za obuku

3. Razvoj testnog slučaja

Faza razvoja testnog slučaja uključuje stvaranje, provjeru i preradu testnih slučajeva i testnih skripti nakon što je plan testiranja spreman. U početku se podaci o testu identificiraju, zatim stvaraju i pregledavaju, a nakon toga ponovno obrađuju na temelju preduvjeta. Naposljetku, tim inženjera osiguranja kvalitete započinje proces razvoja testnih slučajeva za pojedinačne jedinice.

Aktivnosti razvoja testnih slučajeva:

- Stvaranje testnih slučajeva, skripti za automatizaciju (ako je primjenjivo)
- Provjera osnovnih testnih slučajeva i skripti
- Stvaranje testnih podataka (ako je testno okruženje dostupno)

4. Postavljanje testnog okruženja

Postavljanje testnog okruženja odlučuje o softverskim i hardverskim uvjetima pod kojima se radni proizvod testira. To je jedan od kritičnih aspekata procesa testiranja i može se raditi paralelno s fazom razvoja testnog slučaja. Testni tim možda neće biti uključen u ovu aktivnost ako razvojni tim osigurava testno okruženje. Ispitni tim je dužan izvršiti provjeru spremnosti danog testnog okruženja.

Aktivnosti postavljanja testnog okruženja:

- Razumijevanje potrebne arhitekture, postavke okruženja i priprema popisa hardverskih i softverskih zahtjeva za testno okruženje

- Postavljanje testnog okruženja i testnih podataka

5. Izvršenje testa

Čim faza razvoja završi, tim za testiranje je spreman s testnim slučajevima i započinje testiranje. Proces se sastoji od izvođenja testne skripte, održavanja testne skripte i prijavljivanja grešaka. Ako se prijave greške, to se vraća razvojnom timu na ispravak i izvršit će se ponovno testiranje. Greške se evidentiraju kroz dokument u kojem je pisan testni slučaj, dok prijava greške ide kroz sustav za prijavu istih koju poduzeće koristi. Veliki sustavi imaju noćne obrade podataka koje su od velike važnosti za testiranje tako da je važno izvesti testiranje planski i organizirano. Ispitivač fokusirano vrši testiranje dok ispunjava dokumentaciju testnih scenarija napisanog slučaja. Uz dokumentaciju, prati i specifikaciju zahtjeva zadatka kako bi bio siguran da je zadatak razvijen u skladu s istom. Evidentira greške i ako je potrebno prokomentira ih s drugim ispitivačima koji nisu detaljno upućeni u zadatak. Bitno je i napomenuti da ispitivač uzme stanku nakon određenog vremena testiranja, malo se 'odmakne' i predahne od zadatka. U suprotnome, teško će moći ispravno reagirati na nove iteracije testiranja. Nakon što je greška ispravljena te ukoliko nije došlo do izmjene specifikacije, slijedi nova iteracija testiranja istog testnog slučaja te evidencija krajnjeg rezultata. Ukoliko dođe do promjene specifikacije, razmatra se na koje testne slučajeve je utjecala promjena iste, te se vrši nova iteracija testiranja istog testnog slučaja.

Aktivnosti izvršenja testa:

- Provođenje testova prema planu
- Dokumentiranje rezultata testiranja i zapisivanje nedostataka za neuspjele slučajeve
- Ponovno testiranje popravaka kvarova
- Praćenje nedostataka do zatvaranja

6. Zatvaranje ciklusa ispitivanja

Faza zatvaranja testnog ciklusa je završetak izvođenja testa koji uključuje nekoliko aktivnosti kao što su izvješćivanje o završetku testa, prikupljanje matrica završetka

testa i rezultata testa. Članovi tima za testiranje sastaju se, raspravljaju i analiziraju artefakte testiranja kako bi identificirali strategije koje se moraju implementirati u budućnosti, uzimajući u obzir lekcije iz trenutnog ciklusa testiranja.

Aktivnosti zatvaranja ciklusa testiranja:

- Ocjenjivanje kriterija završetka ciklusa na temelju vremena, pokrivenosti testom, cijene, softvera, ključnih poslovnih ciljeva, kvalitete
- Pripremanje metrike testa na temelju gore navedenih parametara
- Dokumentiranje kako bi se moglo ubuduće učiti iz projekta
- Pripremanje izvješća o zatvaranju testa
- Kvalitativno i kvantitativno izvješćivanje kupca o kvaliteti rada proizvoda
- Analiziranje rezultata testa kako bi se saznala raspodjela kvarova prema vrsti i težini

8. Norme i načela testiranja

Primjena normi u testiranju poboljšava kvalitetu softvera, odnosno primjena standardiziranih testnih metoda pomaže u postizanju konzistentnosti i pouzdanosti rezultata testiranja (Myers et al., 2013). Norme testiranja softvera su skup smjernica i načela koji se koriste za definiranje i vođenje procesa testiranja softvera. Oni opisuju strategije, postupke, metodologije i najbolju praksu za testiranje softvera kako bi se osigurala njegova kvaliteta, funkcionalnost i izvedba. Oni također pružaju temelj za kreiranje i evaluaciju procesa, tehnika i alata za testiranje softvera. Neke uobičajene norme za testiranje softvera uključuju ISO/IEC 29119, IEEE 829 i ISO/IEC 9126. Pridržavanje ovih normi može pomoći organizacijama da poboljšaju djelotvornost i učinkovitost svojih procesa testiranja softvera i na kraju isporuče visokokvalitetne softverske proizvode klijentima (Testsigma, 2024).

Norme testiranja softvera važan su dio procesa razvoja softvera. Mogu pomoći u poboljšanju kvalitete softvera, smanjenju rizika od grešaka i poboljšanju učinkovitosti testiranja softvera. Uz testiranje softvera, važno je upoznati se sa normama testiranja softvera koji se primjenjuju na projekt. To nam pomaže da budemo sigurni da softver testiramo učinkovito i da ispunjavamo očekivanja svojih dionika. Iskustvo od početaka informatike je pokazalo da su neki procesi razvoja softvera bili mnogo učinkovitiji od drugih. Kako je softverska industrija rasla, postala je očita potreba za normama unutar discipline softverskog inženjerstva. Mnoge globalne organizacije za norme poput Međunarodne organizacije za standardizaciju (ISO) propisuju norme za poboljšanje kvalitete softvera. Lista normi se nalazi na tablici 2.

Standard	Description
CMMI-Dev	A process improvement model for software development.
TMMI	A process improvement model for software testing.
ISO/IEC/IEEE 29119	A set of standards for software testing.
ISO/IEC 25000:2005	A standard for software product quality requirements and evaluation (SQuaRE).
ISO/IEC 12119	A standard that establishes requirements for software packages and instructions on how to test a software package against those requirements.
IEEE 829	A standard for the format of documents used in different stages of software testing.
IEEE 1061	Defines a methodology for establishing quality requirements, identifying, implementing, analysing, and validating the process and product of software quality metrics.
IEEE 1059	Guide for software verification and validation plans.
IEEE 1008	A standard for unit testing.
IEEE 1012	A standard for software verification and validation.
IEEE 1028	A standard for software inspections.
IEEE 1044	A standard for the classification of software anomalies.
IEEE 1044-1	A guide to the classification of software anomalies.
IEEE 830	A guide for developing system requirements specifications.
IEEE 730	A standard for software quality assurance plans.
IEEE 1061	A standard for software quality metrics and methodology.
IEEE 12207	A standard for software life cycle processes and life cycle data.
BS 7925-1	A vocabulary of terms used in software testing.
BS 7925-2	A standard for software component testing.

Tablica 2 Lista normi, izvor: PeopleCert International Ltd 2023, str. 21

Na primjer, ISO/IEC/IEEE koji ima broj 29119 namijenjen je testiranju softvera i djeluje kao međunarodno odobrena zbirka normi u testiranju softvera koja se slijedi za bilo koji SDLC model u razvoju softvera za bilo koju organizaciju. Kada implementiramo norme, usvajamo međunarodno priznate i odobrene norme testiranja koji će našoj organizaciji na kraju ponuditi kvalitetan pristup testiranju (*PeopleCert International Ltd 2023*).

U testiranju web portala i desktop/mobilne aplikacije koristi se ISO 25000 norma. Testiranje je proces evaluacije isporučenog proizvoda s namjerom pronalaženja pogrešaka. Ono nije samo faza projekta ili samo pronalazak neispravnog koda, završni ispit i otklanjanje pogrešaka. Glavni cilj testiranja je otkriti pogreške koje se mogu pojaviti tijekom razvoja i integracije softverskih komponenti (Sommerville, 2016).

Načela testiranja su važna za stručnjake testiranja jer pružaju temelj za razvoj znanja o testiranju i stjecanje vještina testiranja. Oni također daju smjernice za definiranje aktivnosti testiranja koje se izvode u praksi stručnjaka za testiranje.

Princip se može definirati kao:

1. opći ili temeljni zakon, doktrina ili pretpostavka
2. pravilo ili kodeks ponašanja

3. zakoni ili činjenice prirode na kojima se temelji rad umjetne naprave
(*PeopleCert International Ltd 2023*).

Testiranje od strane pojedinca koji je izradio proizvod nije se pokazalo uspješnim u mnogim organizacijama. Nedostaci osobe koja provjerava vlastiti rad su sljedeći:

- Nesporazumi se neće otkriti jer će programer pretpostaviti da je ono što su čuli točno
- Nepravilno korištenje razvojnog procesa možda neće biti otkriveno jer pojedinac možda ne razumije proces
- Pojedinac može biti "zaslijepljen" u prihvaćanju pogrešnih specifikacija sustava i kodiranja jer upada u istu zamku tijekom testiranja koja je uopće dovela do uvođenja kvara
- Programeri softvera optimistični su u pogledu svoje sposobnosti da rade bez grešaka i stoga ponekad podcjenjuju potrebu za opsežnim testiranjem

Bez formalne podjele između razvoja i testiranja, pojedinac bi mogao biti u iskušenju da poboljša strukturu sustava i dokumentaciju, umjesto da to vrijeme i trud posveti testiranju.

Testiranje softvera razlikuje se od organizacije do organizacije. Mnogi čimbenici utječu na testiranje. Glavni faktori su:

- ljudski odnosi
- opseg testiranja
- razumijevanje vrijednosti testiranja životnog ciklusa
- loše planiranje testiranja
- ograničenja testiranja.

Testovi se mogu klasificirati prema tome jesu li izvedeni iz opisa funkcije programa, iz strukture programa ili iz implementacije atributa kvalitete i karakteristika sustava. I strukturalna, funkcionalna i nefunkcionalna ispitivanja treba provesti kako bi se osiguralo odgovarajuće ispitivanje. Strukturni testni setovi imaju tendenciju otkrivanja grešaka koje se javljaju tijekom kodiranja programa; setovi funkcionalnih testova imaju tendenciju otkrivanja pogrešaka koje se javljaju u implementaciji zahtjeva ili specifikacija dizajna; a nefunkcionalni testovi imaju tendenciju otkrivanja lošeg dizajna i kodiranja (*PeopleCert International Ltd 2023*).

9. Ručno testiranje

Ručno testiranje omogućuje otkrivanje specifičnih grešaka koje automatizacija ne može prepoznati. Testiranje softvera je ključan proces u razvoju aplikacija, a može se provoditi na dva osnovna načina: automatskim i ručnim testiranjem. Automatsko testiranje podrazumijeva upotrebu specijaliziranih alata koji omogućuju brzo i učinkovito provođenje testova s minimalnim ljudskom intervencijom. S druge strane, ručno testiranje se oslanja na ljudske testere koji detaljno ispituju funkcionalnost softvera i korisničko iskustvo. Prilikom ručnog testiranja, aplikacija se doživljava na isti način na koji bi ju doživio i krajnji korisnik. Tester može intuitivno identificirati probleme koje automatizirani alati ne mogu otkriti (Kaner i Bach, 2001). Oba pristupa imaju svoje prednosti i nedostatke te se često koriste u kombinaciji kako bi se osiguralo što potpunije pokrivanje testnih slučajeva i otkrivanje grešaka. Iako automatizacija može značajno ubrzati testiranje, ručno testiranje ostaje neophodno za otkrivanje specifičnih grešaka u složenim scenarijima (Myers et al., 2013). Razvoj softvera vrlo je dinamičan proces sa zahtjevima koji se stalno mijenjaju, a ljudski tester mogu se brzo prilagoditi tim promjenama usvajanjem novih pristupa testiranju.

Ručno testiranje ERP sustava se odnosi na proces gdje inženjeri osiguranja kvalitete ili tester ručno pregledavaju i testiraju funkcionalnosti ERP sustava kako bi osigurali ispravan rad sustava prema definiranim specifikacijama. Za razliku od automatskih testova koji koriste skripte za ponavljanje određenih scenarija, ručno testiranje uključuje stvarnog korisnika koji ručno izvodi korake unutar sustava. U nastavku je prikazano pet ključnih aspekata ručnog testiranja ERP sustava:

1. Testiranje funkcionalnosti:

Naglasak je na provjeri različitih modula ERP sustava (npr. financije, nabava, prodaja) kako bi se osigurala ispravnost funkcioniranja. Tester prate već unaprijed definirane scenarije, poput unosa podataka u sustav, izrade izvještaja ili obrade transakcija.

2. Testiranje integracije:

Budući da ERP sustavi integriraju mnoge poslovne procese, važno je testirati kako međusobno komuniciraju različiti moduli. Ovdje ručno testiranje pomaže u identifikaciji

problema u komunikaciji između modula, kao što su npr. problemi u prijenosu podataka između odjela financija i prodaje ili između odjela nabave i skladišta.

3. Testiranje korisničkog sučelja:

Ručno testiranje korisničkog sučelja omogućuje procjenu koliko je ERP sustav intuitivan za krajnje korisnike. Testerima procjenjuju dizajn, navigaciju i jednostavnost korištenja. Također, identificiraju sve potencijalne nedostatke u korisničkom iskustvu.

4. Testiranje upotrebljivosti:

Testiranjem se procjenjuje koliko je sustav prilagođen krajnjim korisnicima, koliko je jednostavan za korištenje te koliko su korisnici zadovoljni njegovim performansama. Naglasak je na iskustvu korisnika i njihovoj interakciji s ERP-om u stvarnim poslovnim scenarijima.

5. Testiranje performansi:

Ručno testiranje može uključivati i osnovnu procjenu performansi sustava, poput brzine odziva, vremena obrade transakcija i opterećenja.

Prednosti i nedostaci ručnog testiranja

Prednosti ručnog testiranja ERP sustava su fleksibilnost i korisnički uvid. Ručno testiranje omogućuje testerima istražiti aplikaciju na kreativan način, izvan unaprijed definiranih skripti, što može pomoći u otkrivanju neočekivanih problema. Budući da testerima provode korake koje bi korisnici poduzeli, mogu uočiti potencijalne probleme s korisničkim iskustvom, što je teško simulirati u automatiziranom testiranju.

Nedostaci ručnog testiranja ERP sustava su vremenska zahtjevnost i rizik od ljudske pogreške. Ručno testiranje je često sporije od automatiziranog i zahtijeva značajno vrijeme za pokrivanje svih funkcionalnosti. Budući da ručno testiranje provode ljudi, postoji mogućnost za neuočene greške ili nepravilnosti.

Ručno testiranje ERP sustava ima ključnu ulogu u provjeri funkcionalnosti, integracije i upotrebljivosti sustava. Iako zahtijeva više vremena i resursa, važno je za osiguranje kvalitete ERP sustava, osobito u ranim fazama razvoja ili kada se unose promjene koje trebaju brzu i temeljitu provjeru. Iako automatizacija testiranja donosi mnoge prednosti, ručno testiranje ostaje nezamjenjivo za određene vrste složenih scenarija (Myers et al., 2013).

9.1 Prednosti ručnog testiranja u odnosu na automatsko

Ručno testiranje pronalazi sve vrste grešaka i problema, kao i automatsko testiranje. Međutim, postoje neki problemi u softveru kod kojih se ručno testiranje ističe u otkrivanju gdje bi automatizacija propustila.

Loš tijek rada

Tijek rada odnosi se na put koji korisnik slijedi kako bi došao do određene točke u aplikaciji i dovršio proces. Iako možda nema ničeg tehnički pogrešnog s nekim tijekovima rada, oni ipak mogu biti problematični jer put možda nema smisla manje iskusnom korisniku. U tim slučajevima ručni tester obavještava razvojnog programera o problemima s dizajnom i preporuča promjene, pomažući korisnicima da budu bolje upoznati s aplikacijom na način koji automatski sustavi ne bi shvatili.

Grafički problemi

Web aplikacije rade na nizu uređaja, pri čemu rezolucije i veličine monitora stalno variraju ovisno o telefonu, tabletu ili zaslonu koji korisnik ima na raspolaganju. U loše optimiziranoj aplikaciji to bi moglo dovesti do toga da sredstva postanu rastegnuta i izgledaju lošije na uređajima koji se rjeđe koriste, s alatima za automatizaciju koji jednostavno slijede izbornike i ne primjećuju to. Implementacijom niza uređaja, ručni tester pronađe grafičke nedostatke koji, kada se isprave, uzrokuju bolje iskustvo korisnika sa softverskim paketom.

Netočne poveznice

Neke web stranice ili aplikacije su povezane s web stranicama društvenih medija putem niza gumba i ugrađenih veza. Međutim, oni se možda neće uvijek povezati s pravim mjestom kao rezultat tipfelera ili pogreške u procesu razvoja, što je nešto što automatski sustav neće nužno pronaći. Veze koje idu na krivo mjesto mogu izazvati zabunu i značajno naštetiti zadržavanju. Ručni tester prolazi kroz sve poveznice u programu i osigurava da one vode na pravo mjesto, pomažući krajnjim korisnicima da stignu kamo žele, a ne da budu zavedeni problemom.

9.2 Uobičajene metrike ručnih testera

Mjerni podaci su jednostavne i mjerljive numeričke vrijednosti koje pokazuju rezultate nakon završetka testa. Najčešće su kvantitativne prirode, što ih čini lakšim za procjenu iz perspektive programera. Stručnjaci za osiguranje kvalitete provode ručno testiranje

korak po korak. Okviri, alati i softver za automatska testiranja se koriste za izvođenje testova u automatskom testiranju. Ručno testiranje je dugotrajna tehnika, ali testerima omogućuje nositi se sa kompliciranijim okolnostima. Postoje dvije vrste metrika ručnog testiranja:

1. Osnovna metrika: Analitičari prikupljaju podatke tijekom razvoja i izvođenja testnih slučajeva kako bi pružili osnovnu metriku. Generiranjem izvješća o statusu projekta, te se metrike šalju voditeljima testiranja i voditeljima projekta. Kvantificira se pomoću izračunatih metrika.
 - Ukupan broj testnih slučajeva
 - Ukupan broj dovršenih testnih slučajeva.
2. Izračunata metrika: Podaci iz osnovne metrike koriste se za izradu izračunate metrike. Voditelj testiranja prikuplja te informacije i pretvara ih u korisnije informacije za praćenje napretka projekta na razini modula, testera i drugim razinama. To je važan aspekt SDLC-a budući da programerima omogućuje kritične promjene softvera.

Druge važne metrike:

- Mjerni podaci o greškama: mjerni podaci o greškama pomažu inženjerima razumjeti mnoge aspekte kvalitete softvera, kao što su funkcionalnost, izvedba, stabilnost instalacije, upotrebljivost, kompatibilnost i tako dalje.
- Pridržavanje rasporeda: glavna svrha pridržavanja rasporeda je određivanje vremenske razlike između očekivanog i stvarnog vremena izvršenja rasporeda.
- Ozbiljnost kvara: ozbiljnost problema omogućuje programeru vidjeti kako će kvar utjecati na kvalitetu softvera.
- Učinkovitost testnog slučaja: učinkovitost testnog slučaja mjera je koliko su testni slučajevi učinkoviti u otkrivanju problema.
- Stopa pronalazjenja nedostataka: koristi se za određivanje uzorka nedostataka tijekom određenog vremenskog razdoblja.
- Vrijeme otklanjanja kvara: vrijeme potrebno za otklanjanje problema poznato je kao vrijeme otklanjanja kvara.
- Testna pokrivenost: određuje broj testnih slučajeva dodijeljenih programu. Ova metrika osigurava da je testiranje potpuno dovršeno. Također pomaže u provjeri protoka koda i testiranju funkcionalnosti.

- Uzrok kvara: koristi se za utvrditi što uzrokuje problem.

Testne metrike različitih faza životnog ciklusa testiranja su analiza, komunikacija, evaluacija i izvješće. Testne metrike moraju biti identificirane i prepoznate u fazi analize životnog ciklusa testiranja. Nakon prepoznavanja, potrebno je definirati metriku osiguranja kvalitete koja je relevantna za određeni projekt. Sljedeći korak odnosi se na komunikaciju, gdje je važno obavijestiti sve zainteresirane strane i tim za testiranje o potrebi prikupljanja metričkih podataka. Istovremeno, tim za testiranje treba educirati o tome koje podatke treba prikupiti kako bi se osigurala točna obrada metričkih informacija. U fazi evaluacije, podaci se moraju prikupiti, a zatim provjeriti njihova točnost. Prikupljeni podaci koriste se za izračunavanje vrijednosti metrike, čime se osigurava da su svi relevantni pokazatelji pravilno evaluirani. Nakon toga slijedi izrada izvješća, gdje je važno donijeti zaključke na temelju prikupljenih i analiziranih podataka. Na kraju, izvješće treba podijeliti s odgovarajućim dionicima i predstavnicima, te prikupiti povratne informacije od njih kako bi se osigurala daljnja optimizacija procesa (Hamilton, 2024).

Mjerni podaci korišteni u empirijskom dijelu rada, testiranju korisničkog portala i mobilne aplikacije, temelje se na standardnim metrikama za evaluaciju softverskih sustava. Ove metrike omogućuju procjenu različitih aspekata funkcionalnosti, stabilnosti i upotrebljivosti sustava. Metrike koje su bile primijenjene u tri studije slučaja obuhvaćaju sljedeće kategorije:

1. Osnovna metrika

Ukupan broj testnih slučajeva: ova metrika kvantificira ukupan broj kreiranih i provedenih testnih slučajeva u sklopu testiranja. Koristi se za praćenje obuhvata testiranja.

Ukupan broj dovršenih testnih slučajeva: prati broj uspješno dovršenih testnih slučajeva, što osigurava podatke o napretku testiranja i dovršenosti ispitivanja funkcionalnosti.

2. Izračunata metrika

Podaci prikupljeni iz osnovne metrike koriste se za izračun naprednijih indikatora uspješnosti, uključujući:

Postotak dovršenih testnih slučajeva: Mjeri omjer dovršenih testnih slučajeva u odnosu na ukupan broj testova.

Učinkovitost testiranja: Ova metrika kvantificira učinkovitost pronalaženja i rješavanja grešaka u softverskom sustavu.

3. Mjerni podaci o greškama

Ove metrike pomažu pri evaluaciji kvalitete softvera, uključujući:

Stopa otkrivenih grešaka: Postotak grešaka otkrivenih tijekom provođenja testnih slučajeva.

Stopa popravljenih grešaka: Omjer popravljenih grešaka u odnosu na ukupan broj otkrivenih grešaka.

Vrijeme otklanjanja grešaka: Vrijeme potrebno za otklanjanje svake prijavljene greške, što daje uvid u brzinu reakcije tima za razvoj.

4. Pridržavanje rasporeda

Ova metrika prati vremensku razliku između planiranih i stvarnih vremena za izvršenje pojedinih zadataka i testnih slučajeva.

5. Ozbiljnost kvara

Ozbiljnost svakog kvara ili greške omogućuje procjenu utjecaja pojedinog problema na ukupnu funkcionalnost sustava, što programerima pruža uvid u prioritet rješavanja problema.

6. Testna pokrivenost

Testna pokrivenost mjeri koliko je testnih slučajeva dodijeljeno pojedinom modulu ili funkcionalnosti sustava, čime se osigurava da je testiranje sustava sveobuhvatno.

7. Stopa novih zahtjeva

U sklopu evaluacije, pratila se i stopa novih funkcionalnih zahtjeva koji su identificirani tijekom testiranja, s ciljem prilagodbe sustava potrebama korisnika.

8. Stopa prihvaćenih nedostataka

Ova metrika mjeri omjer nedostataka koji su prihvaćeni za rješavanje od strane razvojnog tima u odnosu na sve prijavljene nedostatke, čime se osigurava povratna informacija o kvaliteti otkrivenih problema.

U empirijskom dijelu rada korištena je kombinacija metrika, koje su prilagođene prema specifičnostima svake studije slučaja, kako bi se procijenila učinkovitost, upotrebljivost i kvaliteta testiranog sustava. Testiranje korisničkog portala temeljilo se na detaljnim metrikama upotrebljivosti, dok su kod mobilne i desktop aplikacije dodatno naglašene metrike performansi i prilagodljivosti. U konkretnom slučaju, odabrane su metode prema ISO/IEC 25062:2006 (CIF) normi jer omogućuju strukturiranu evaluaciju korisničkog sučelja i funkcionalnosti iz perspektive krajnjih korisnika. Ove metode su usmjerene na korisničko iskustvo i omogućuju analizu korisničke percepcije softvera u stvarnim uvjetima korištenja. Osim toga, CIF norma pruža jasno definirane kriterije koji se lako mogu primijeniti u procesu heurističke evaluacije, čime se omogućuje rano prepoznavanje problema u dizajnu i funkcionalnosti ERP sustava.

Kaner i Bach naglašavaju važnost korištenja metoda usmjerenih na korisnike jer omogućuju detaljan uvid u stvarne interakcije korisnika sa sustavom i osiguravaju pravovremenu identifikaciju upotrebljivih rješenja koja poboljšavaju korisničko iskustvo (Kaner & Bach, 2001). Stoga je odabir ove norme opravdan njegovom širokom primjenom i relevantnošću za složene softverske sustave, poput ERP-a, koji zahtijevaju jednostavnost korištenja i visoku razinu intuitivnosti zbog velikog broja korisnika različitih tehničkih profila.

9.3 Vrste testiranja koje se izvode ručno

Vrste testiranja koje se izvode ručno obuhvaćaju različite metode provjere funkcionalnosti, upotrebljivosti i performansi softverskog sustava, a svaka od tih metoda ima specifičan cilj u osiguravanju kvalitete prije isporuke krajnjim korisnicima. Bilo koja vrsta testiranja može se izvršiti ručno ili uz pomoć alata za automatizaciju, a konačna odluka o odabiru pristupa leži u prirodi tog specifičnog testnog slučaja. Nekoliko vrsta testiranja koje se obično izvode ručno prikazane su u nastavku.

Testiranje prihvaćanja korisnika

Ispitivanje prihvaćanja od strane korisnika uključuje provjeru je li proizvod prikladan za namjeravanu bazu klijenata proizvoda. To uključuje omogućavanje potencijalnim klijentima pristupa aplikaciji kako bi je mogli koristiti i dati povratne informacije (Testsigma, 2024). Jedan od najčešćih primjera testiranja prihvaćanja od strane korisnika u modernom razvoju softvera je alfa i beta testiranje videoigara, u kojem

igrači igraju igru i izvješćuju o svim problemima koji postoje u njoj. Glavna prednost dovršetka testiranja prihvaćanja od strane korisnika je to što dobivate vanjsku perspektivu o svom proizvodu umjesto da se oslanjate na perspektivu ljudi koji su imali aktivnu ulogu u stvaranju proizvoda, čime se uklanja svaki potencijal za pristranost koja utječe na testiranje. Ručno testiranje je neophodno jer sustav automatizacije ne može točno preslikati mišljenje korisnika.

Istraživačko testiranje

Tester istražuju aplikaciju bez unaprijed definiranih testnih slučajeva. Cilj im je otkriti nepoznate probleme i ponašanja sustava kroz kreativno i intuitivno istraživanje. Ovakav pristup naglašava koliko su tester stručni, iskusni i sposobni razmišljati izvan okvira. Istraživačko testiranje softvera je posebno korisno u ranim fazama razvoja kada je važno brzo identificirati ključne probleme i dobiti općeniti osjećaj za kvalitetu aplikacije. Također se može koristiti kao dodatak formalnim metodama testiranja kako bi se osigurala šira pokrivenost i kako bi se otkrili problemi koji su možda propušteni tijekom formalnog testiranja.

Testiranje upotrebljivosti

Tester istražuju koliko je softverski proizvod jednostavan za korištenje, intuitivan i pristupačan krajnjim korisnicima. Cilj je osigurati da korisnici mogu efikasno, efektivno i zadovoljno koristiti softver za obavljanje svojih zadataka. Ovo testiranje se često provodi uz pomoć stvarnih korisnika koji izvršavaju zadatke dok ih tester promatra i bilježi njihova iskustva. Ručno testiranje upotrebljivosti je ključni korak u osiguravanju da je softver intuitivan i da zadovoljava potrebe krajnjih korisnika.

Funkcionalno testiranje

Tester kvalitete istražuju rade li značajke testirane aplikacije u skladu s njihovim specificiranim zahtjevima. Može se izvesti ili ručno ili pomoću automatiziranih alata na temelju specifičnih testnih slučajeva. Nekoliko primjera funkcionalnih testnih slučajeva uključuje:

- Provjerite uspješnu prijavu s valjanim vjerodajnicama
- Provjerite ponašanje sustava za prijavu s nevažećim vjerodajnicama
- Provjerite funkciju pretraživanja proizvoda
- Provjerite protok podataka u pozadini

Primarni cilj funkcionalnog testiranja je osigurati da softver obavlja funkcije za koje je dodijeljen, a obično se ne bavi internom strukturom koda ili detaljima implementacije (Katalon, 2024).

Nefunkcionalno testiranje

Nefunkcionalno ručno testiranje softvera odnosi se na testiranje aspekata sustava koji nisu izravno povezani s određenim funkcijama ili značajkama softvera. Umjesto toga, fokusira se na karakteristike sustava kao što su performanse, sigurnost, upotrebljivost, pouzdanost i kompatibilnost. Nefunkcionalno ručno testiranje je ključni dio osiguranja kvalitete softvera jer pomaže identificirati i riješiti probleme koji mogu utjecati na korisničko iskustvo i ukupnu pouzdanost sustava.

10. Validacija upotrebljivosti ERP proizvoda

Ovo poglavlje je posvećeno empirijskom dijelu rada, prikazuje vrednovanje i testiranje provedene validacije upotrebljivosti ERP proizvoda. Cilj istraživanja je potvrditi korisničku upotrebljivost ERP proizvoda, tj. zadovoljava li specifične potrebe korisnika u stvarnom poslovnom okruženju, te da su dizajn i funkcionalnost sustava intuitivni, učinkoviti i prilagođeni krajnjim korisnicima. ERP sustav je ručno testiran kroz testiranje upotrebljivosti, uz heurističko vrednovanje, stoga je fokus validacije je na sljedećim ciljevima.

1. Potvrditi učinkovitost ERP sustava iz perspektive korisničkog iskustva (UX)

Validacija provjerava hoće li krajnji korisnici lako i intuitivno koristiti ERP sustav u stvarnim uvjetima. To uključuje provjeru funkcionalnosti u odnosu na uobičajene zadatke i scenarije koje korisnici obavljaju u svakodnevnom radu.

2. Procijeniti prilagodljivost sustava različitim korisničkim profilima

Sustav se testira s različitim skupinama korisnika (npr. administratori, zaposlenici, menadžeri) kako bi se utvrdilo je li jednostavan za upotrebu neovisno o tehničkoj stručnosti korisnika. Cilj je osigurati da svi profili korisnika mogu uspješno i učinkovito obavljati svoje zadatke.

3. Potvrditi rezultate heurističke evaluacije

Heuristička evaluacija je tehnika koja se koristi za identificiranje potencijalnih problema u dizajnu korisničkog sučelja na temelju skupine unaprijed definiranih pravila ili smjernica (heuristika). Validacija potvrđuje da su problemi identificirani heurističkom evaluacijom ispravljani i da su implementirana rješenja za povećanje upotrebljivosti bila uspješna.

4. Potvrditi zadovoljstvo korisnika

Kroz validaciju se procjenjuje koliko su korisnici zadovoljni načinom na koji ERP sustav olakšava ili podržava njihove zadatke, smanjuje greške i omogućuje brže izvršenje poslovnih procesa. Validacija se također bavi smanjenjem frustracija kod korisnika, posebno ako je tijekom testiranja uočeno da je neka funkcionalnost bila nejasna ili komplicirana.

5. Osigurati efikasnost u stvarnim uvjetima rada

Cilj je utvrditi kako ERP sustav radi u poslovnom okruženju, s pravim podacima i radnim uvjetima. Validacija provjerava hoće li sustav, kad ga koriste stvarni korisnici, zaista povećati produktivnost i efikasnost rada u organizaciji. Validacija upotrebljivosti ERP proizvoda osigurava da sustav nije samo tehnički ispravan (što je cilj testiranja), već i upotrebljiv, intuitivan i efikasan za krajnje korisnike u stvarnim uvjetima. To potvrđuje da su svi aspekti korisničkog iskustva optimizirani te da ERP sustav donosi stvarnu poslovnu vrijednost kroz povećanje efikasnosti i zadovoljstva korisnika.

Testiranje web Korisničkog portala, desktop verzije aplikacije iDoctor i mobilne verzije aplikacije mVizita je provedeno prema metrikama norme ISO/IEC25062:2006 te prema knjizi "Metrics and Models in Software Quality Engineering" autora S. H. Kana. CIF norma definira formate i metodologije za izvještavanje o testiranju upotrebljivosti. Norma objašnjava kako kvantificirati metrike vezane uz korisničko iskustvo, poput vremena izvršenja zadataka i omjera uspjeha i neuspjeha optimizacija, kao i kako interpretirati rezultate testiranja u kontekstu poboljšanja softverskih sustava. U svojoj knjizi, Stephen H. Kan objašnjava širok raspon metrika za procjenu performansi softverskih sustava, uključujući mjerenje razlike u performansama prije i nakon optimizacije, kao i analizu uspješnosti softverskih optimizacija.

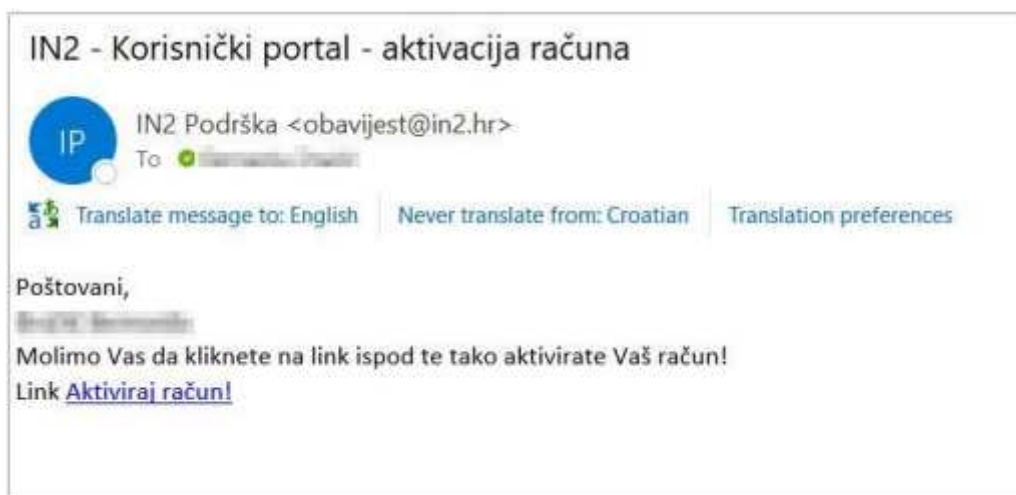
10.1 Validacija upotrebljivosti web portala

U ovom poglavlju je predstavljeno korištenje korisničkog portala te su prikazani rezultati studije slučaja testiranja Korisničkog portala. Međunarodna organizacija za standardizaciju (eng. the International Organization for Standardization) i Međunarodna elektrotehnička komisija (eng. the International Electrotechnical Commission) čine specijalizirani sustav za svjetsku standardizaciju. Nacionalna tijela koja su članovi ISO-a ili IEC-a sudjeluju u razvoju međunarodnih normi putem tehničkih odbora koje je osnovala odgovarajuća organizacija za bavljenje određenim područjima tehničke djelatnosti. ISO i IEC tehnički odbori surađuju u područjima od zajedničkog interesa. U radu sudjeluju i druge međunarodne organizacije, vladine i nevladine, u vezi s ISO-om i IEC-om. U području informacijske tehnologije, ISO i IEC uspostavili su zajednički tehnički odbor, ISO/IEC JTC 1. Međunarodne norme izrađene su u skladu s pravilima danim u ISO/IEC Direktivama.

Odabrana norma za ručno testiranje web portala je ISO/IEC 25062:2006, poznata kao Uobičajeni industrijski format (CIF) za izvješća o ispitivanju upotrebljivosti (eng. Common Industry Format (CIF) for Usability Test Reports). Iako je ova norma specifično usmjerena na dokumentiranje rezultata upotrebljivosti, on pruža detaljan okvir za provođenje i dokumentiranje ručnih testova usmjerenih na upotrebljivost.

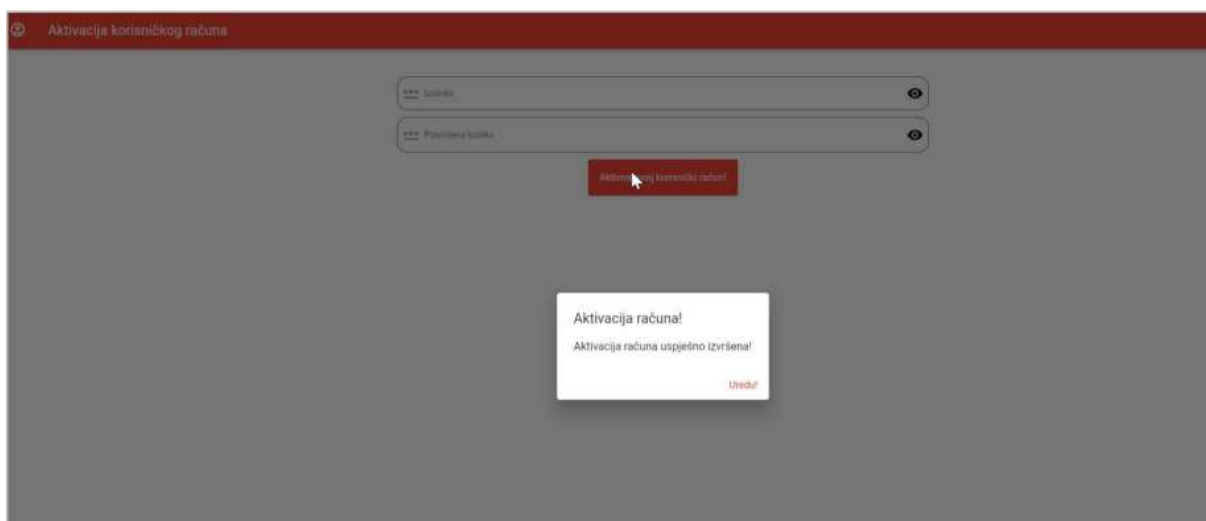
Korisnički portal je zamišljen kao sučelje i mjesto za kolaboraciju sa klijentima pri čemu je klijent ustanova s kojom IN2 ima ugovor za implementaciju/održavanje nekog od proizvoda iz IBIS portfelja. Korisnik portala je ovlaštena osoba od strane Klijenta koja ima dozvolu prijave grešaka prema IN2.

Nakon što je korisniku omogućeno korištenje portala, neovisno o tome koju ulogu ima (administrator/običan korisnik), dobit će e-mail “IN2 - Korisnički portal – aktivacija računa.”



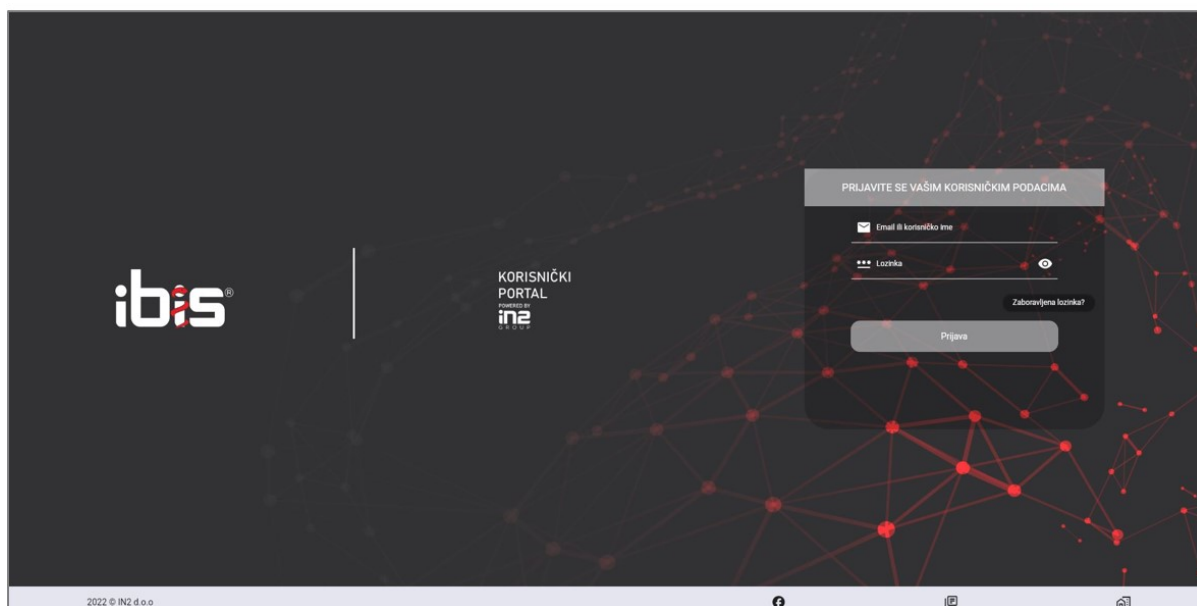
Slika 2 Aktivacija računa, izvor: Korisnički portal IN2 grupe

Pritiskom na link „Aktiviraj račun!“, otvara se sučelje u kojem korisnik unosi željenu lozinku koju će koristiti za prijavu u korisnički portal.



Slika 3 Aktivan račun, izvor: Korisnički portal IN2 grupe

Nakon unosa lozinke i pritiska na gumb [Aktiviraj korisnički račun], javlja se poruka o uspješnoj aktivaciji računa te je korisnik automatski preusmjeren na početnu stranicu portala.

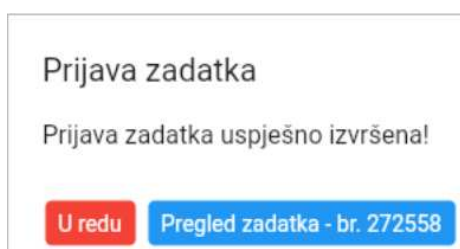


Slika 4 Login sučelje, izvor: Korisnički portal IN2 grupe

Nakon uspješne aktivacije računa i uspješne prijave u sučelje, otvara se prozor "Prijavljivanje zadataka" gdje korisnik može započeti s procesom prijave zadatka.

U formi prijavljivanja zadatka, korisnik može odabrati i kategoriju, prioritet, proizvod, te domenu. Omogućeno je i dodavanje priloga koji se vežu uz prijavu putem gumba [Dodavanje priloga] čijim se odabirom otvara prozor za odabir željenog priloga. Nakon

unosu željenih podataka, prijava se podnosi putem gumba [Prijava] nakon čega se otvara prozor koji korisnika obavještava o uspješnom izvršavanju prijave. Pritiskom na [U redu], korisnik ostaje u čvoru „Pregled zadataka“ gdje je bio preusmjeren nakon prijave zadatka. Pritiskom na gumb za pregled prijavljenog zadatka otvara se detaljni pregled prijave.



Slika 5 Obavijest o prijavi zadatka, izvor: Korisnički portal IN2 grupe

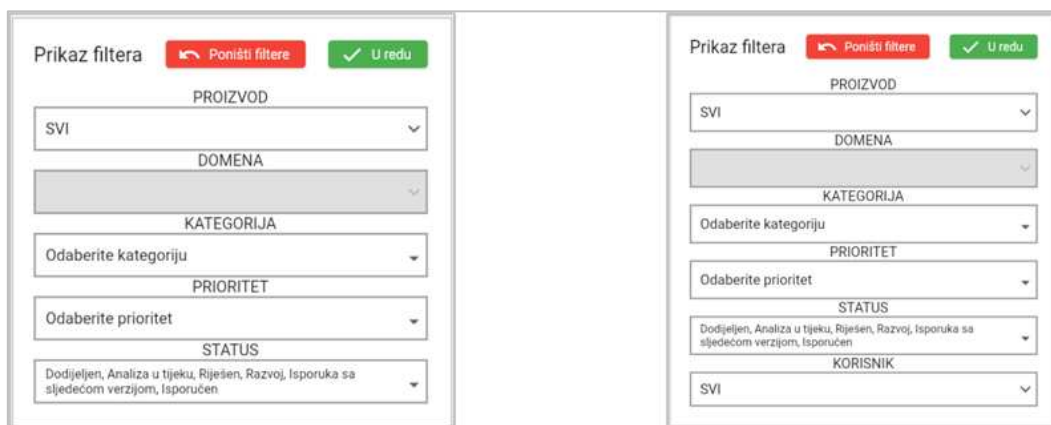
Odabirom čvora „Pregled zadatka“ u izborniku s lijeve strane sučelja, otvara se prozor za pregled unesenih zadataka. Ukoliko korisnik ima ulogu Administrator, u pregledu zadataka će vidjeti zadatke svih korisnika svoje ustanove. Ukoliko korisnik ima ulogu Običan korisnik, u pregledu zadataka će vidjeti samo svoje prijave. U gornjem dijelu sučelja nalazi se dio za pretragu prijave prema filterima. Filteri koji se uvijek prikazuju su pretraga zadataka, vremensko razdoblje i broj zadatka.

Status zadatka	Broj zadatka	Sažetak	Kategorija	Proizvod	Domena	Prioritet	Vrijeme upisa
Dodjeljen	272307	Test prijava 14	Novi zahtjev	BIS	Fizikalna terapija	Hitno	18.12.2023 09:58
Dodjeljen	272306	Test prijava 13	Novi zahtjev	BIS	Fizikalna terapija	Visoki	18.12.2023 09:57
Dodjeljen	272305	Test prijava 12	Greška	BIS	Izlještajni sustav	Niski	18.12.2023 09:57
Dodjeljen	272304	Test prijava 11	Podrška	BIS	Citologija, patologija	Visoki	18.12.2023 09:57
Dodjeljen	272303	Test prijava 10	Greška	BIS	Dijetetičar	Niski	18.12.2023 09:56
Dodjeljen	272302	Test prijava 9	Novi zahtjev	PIS	PIS Štut	Hitno	18.12.2023 09:53
Dodjeljen	272301	Test prijava 8	Greška	BIS	Poliklinika	Ništa	18.12.2023 09:53
Dodjeljen	272300	Test prijava 7	Greška	PIS	PIS Kraljica	Ništa	18.12.2023 09:53
Dodjeljen	272299	Test prijava 6	Greška	PIS	Ljekarna	Ništa	18.12.2023 09:52
Dodjeljen	272298	Test prijava 5	Podrška	BIS	Radne lute	Niski	18.12.2023 09:52
Dodjeljen	272297	Test prijava 4	Novi zahtjev	BIS	Stara MD	Hitno	18.12.2023 09:52
Dodjeljen	272296	Test prijava 3	Greška	BIS	Terapija	Ništa	18.12.2023 09:51
Dodjeljen	272295	Test prijava 2	Novi zahtjev	BIS	Mikrobiologija	Niski	18.12.2023 09:51
Dodjeljen	272294	Test prijava 1	Greška	BIS	Dijetetičar	Visoki	18.12.2023 09:51

Slika 6 Pregled zadataka, izvor: Korisnički portal IN2 grupe

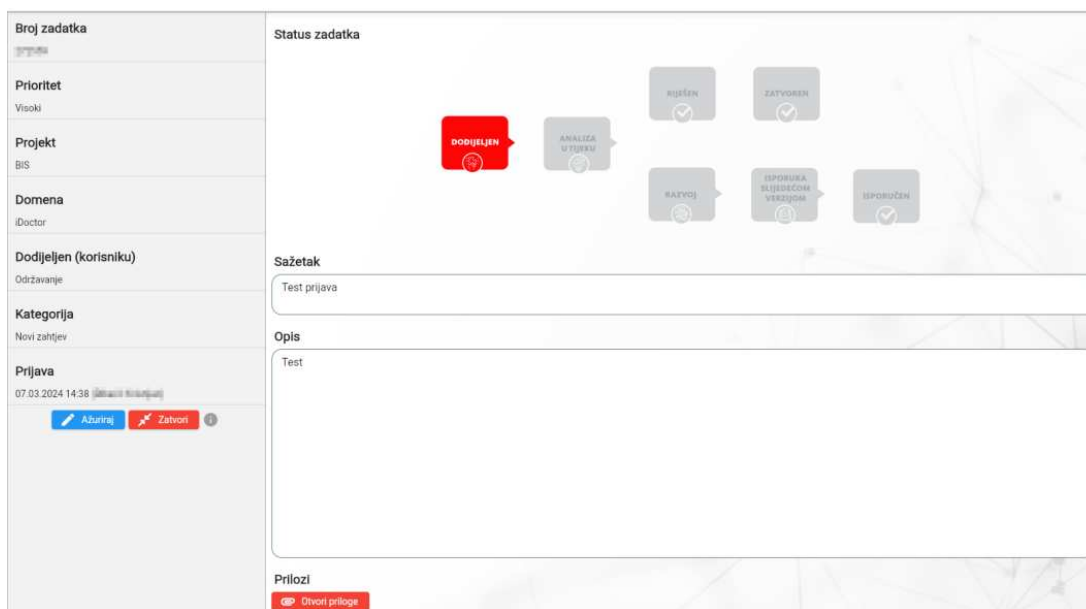
Pritiskom na gumb [Filtriraj] otvaraju se dodatni filteri za pretragu zadataka: proizvod, domena, kategorija, prioritet i status. Dodatno, za ulogu Administrator dostupan je i

filter 'Korisnik.' Za domenu je inicijalno onemogućen odabir vrijednosti jer je vezana za proizvod. Nakon odabira vrijednosti u filteru proizvod, moguće je odabrati i domenu. U filteru status inicijalno su odabrani svi statusi osim statusa 'Zatvoren' kako se u pregledu zadataka ne bi prikazivali zatvoreni zadaci. Prva slika u nastavku prikazuje filter iz pogleda Običnog korisnika, dok druga prikazuje filter iz pogleda Administratora.



Slika 7 Običan korisnik i Administrator, izvor: Korisnički portal IN2 grupe

Odabirom pojedinog zadatka u tabu “Pregled zadataka” otvara se detaljan pregled zadatka s osnovnim informacijama.



Slika 8 Prikaz statusa zadatka, izvor: Korisnički portal IN2 grupe

Nakon što je zadatak preuzet, taj podatak će se ažurirati u polju “Dodijeljen (korisniku).”



Slika 9 Prikaz statusa zatvorenog zadatka, izvor: Korisnički portal IN2 grupe

Po zatvaranju, status zadatka je vidljiv korisniku na prikazu statusa procesa.

Priprema za ručno testiranje web portala prema priručniku norme se prikazuje u nastavku.

- *„Postavite jasne ciljeve za što želite postići testiranjem dizajna web portala, poput identifikacije problema s upotrebljivošću, procjene korisničkog zadovoljstva ili mjerenja efikasnosti navigacije.“*

U testiranju se želi postići provjera upotrebljivosti web portala tj. je li u skladu s korisničkim zahtjevima.

- *„Identificirajte i regrutirajte reprezentativne korisnike koji odgovaraju vašoj ciljanoj publici. Važno je uključiti korisnike s različitim razinama iskustva i vještina.“*

U testiranju su uključeni svi korisnici ustanova koji prijavljuju greške u radu, kao i interni stručnjaci.

- *„Razvijte realistične scenarije i zadatke koje će korisnici izvršavati tijekom testiranja. Ovi zadaci trebaju biti relevantni za stvarne aktivnosti koje korisnici obavljaju na web portalu.“*

U testiranju su uključeni realistični scenariji prijavljivanja grešaka proizvoda. Korisnicima je omogućena faza praćenja cijelog procesa, od prijave do isporuke zadatka. Tijekom internog testiranja se interni stručnjaci ponašaju kao krajnji korisnici, bilježe metrike koje dalje prosljeđuju razvojnom timu.

Provođenje ručnog testiranja prikazano je kroz sljedeće korake.

- *„Prije početka testiranja, pružite korisnicima osnovne informacije o testiranju, uključujući ciljeve testiranja i upute za zadatke koje će izvršavati.“*

Nakon internog testiranja proizvoda korisnicima je održana edukacija o korištenju proizvoda. Već pri edukaciji, korisnici su mogli steći dojam o korištenju iste. Nakon edukacije, korisnici su dobili upute o korištenju. Testovi se prvo izvode na testnoj bazi proizvođača prema heurističkim evaluacijama, te se nakon implementacije, u dogovoru s korisnikom zajedno s njime prolazi određeni test.

- *„Tijekom izvođenja zadataka, promatrajte korisnike i bilježite njihovo ponašanje, komentare i poteškoće koje imaju.“*

Korisniku je dano na odabir da se javi podršci kako bi zajedno prolazili kroz modul. Većina korisnika je bila zadovoljna korištenjem portala te je u rijetkosti bilo potrebe za pozivom tj. spajanjem kod korisnika na računalo.

- *„Mjerite vrijeme potrebno za izvršenje zadatka, broj pogrešaka i stopu uspješnosti zadatka.“*

Vrijeme izvršavanja testiranja zadataka se mjerilo individualno po korisniku jer je krajnji rezultat testa uistinu isporuka ispravka korisniku u sustav. Tako da je vrijeme isporuke ovisilo o riješenim zadacima razvoja. Vrijeme isporuke zadataka je najčešće u dogovoru s ovlaštenom osobom klijenta i vrijeme završavanja procesa nije bilo jedan od zahtjeva korisnika.

- *„Bilježite komentare korisnika (kvalitativne podatke), njihove reakcije i opažanja o dizajnu i funkcionalnosti web portala.“*

Svi komentari korisnika i njihova opažanja su bilježeni u klasičnoj prijavi zadatka u koju konzultant bilježi vrijeme provedeno s korisnikom. Kada od strane korisnika dođe prijava greške o kojoj su komentirali i došli do rješenja, ti zadaci se povezuju i šalju na razvoj programeru koji je na projektu portala na izradu.

- *„Upitnici i intervjui: Nakon testiranja, koristite upitnike i intervjue kako biste dobili povratne informacije o korisničkom iskustvu i zadovoljstvu.“*

S korisnicima je kroz verbalnu komunikaciju dobivena povratna informacija zadovoljstva web portala koja je zadovoljavajuća.

- *„Analiza podataka: Analizirajte prikupljene podatke kako biste identificirali ključne probleme s upotrebljivošću, trendove i područja za poboljšanje.“*

Kvantitativni podaci mogu se analizirati statistički, dok se kvalitativni podaci tematski analiziraju.“

Prikupljeni podaci iz prijave grešaka su analizirani zajedno s provedenim testovima od strane testera. Pri svakom sljedećem testiranju, uvažavaju se i prijašnje prijavljene greške koje se regresijski testiraju.

U nastavku su opisane smjernice struktura izvještaja. ISO/IEC 25062:2006 pruža smjernice za strukturiranje izvještaja koji bi trebao uključivati sažetak, metodologiju, nalaze i preporuke.

- Sažetak: Kratak pregled ciljeva, metodologije, ključnih nalaza i preporuka.

Testiranjem je postignut cilj provjere upotrebljivosti web portala te je u skladu s korisničkim zahtjevima. Uočena je potreba za optimizacijom prikaza podataka koja je naposljetku ispravljena.

- Metodologija: Detalji o korisnicima, zadacima, prikupljenim podacima i analitičkim metodama.

Korisnici su ovlaštene osobe od strane klijenta koji prijavljuju greške proizvoda. Želeći uvid u tijek izrade i isporuke prijavljenog zadatka to je i omogućeno. Nakon što su korisnici prošli prijavu u sustav i prijavili grešku, konzultant je tu prijavu zadatka prosljedio razvojnom timu. Nakon što je zadatak dodijeljen razvojnom timu, korisnik je dalje mogao krenuti s testiranjem. Ovisno o svakom statusu prijavljenog zadatka je ovisilo i vrijeme testiranja. Za to vrijeme neki korisnici su komunicirali s konzultantima i razmjenjivali mišljenja koja su bilježena u zadatke.

- Nalazi: Detaljni opis pronađenih problema s upotrebljivošću, statistički rezultati i kvalitativna opažanja.

Najveći fokus je stavljen na optimizaciju učitavanja podataka. Kako se baza počela puniti podacima, individualno kod svakog korisnika, uočena je usporenost prikaza učitavanja podataka te daljnji rad. Rezultati testiranja su bilježeni tablično. Internim testiranjem se greška brzo riješila. Na svakog korisnika koji koristi portal se gledalo individualno uzimajući u obzir okolinu u kojoj radi (misleći na opterećenje sustava).

Općeniti dizajn portala je u skladu s korisničkim zahtjevima, ali je bilo potrebno prilagoditi responzivnost portala te neke funkcionalne greške (neispravni rad gumba, neispravni prikaz zadataka u razdoblju..).

- Preporuke: Specifične preporuke za poboljšanje upotrebljivosti temeljem nalaza.

Preporuka je uvođenjem metrike koje se prate po korisniku. U takvim slučajevima tester se spajaju kod korisnika te najčešće mjere brzinu nekih određenih aktivnosti (brzina učitavanja stranice, podataka, rada filtera i dr.)

- Implementacija i praćenje

Nakon implementacije proizvoda, prijavom zadataka se pratio broj prijavljenih zadataka putem portala te broj prijavljenih zadataka po bolnicama i mjesecima kroz godine.

- Praćenje i ponavljanje testiranja: Nakon implementacije poboljšanja, ponovite korisničko testiranje kako biste osigurali da su problemi riješeni i da je upotrebljivost poboljšana.

Nakon implementacije poboljšanja ponovljeno je korisničko testiranje u kojem je zabilježeno uspješno otklanjanje grešaka. Nakon svakog ispravka greške, interni stručnjaci provode regresijsko testiranje na korisničkom profilu. Koristeći ISO/IEC 25062:2006 normu osigurava se da je ručno testiranje dizajna web portala temeljito, strukturirano i da pruža vrijedne povratne informacije za poboljšanje upotrebljivosti i ukupnog korisničkog iskustva.

U nastavku su prikazani ciljevi testa s uključenim svim područjima od posebnog interesa.

1. Testiranje korisničke izvedbe proizvoda

- Opis cilja: Cilj je provjeriti mogu li korisnici izvršiti proces prijave zadatka i odgovara li proizvod korisničkim zahtjevima. To uključuje broj grešaka koje naprave, koliko ih uspješno dovrši zadatak i koje promjene žele.
- Kriteriji uspjeha: Proizvod je uspješan ako korisnici mogu obaviti zadatke u razumnom vremenu, s malo ili bez grešaka.

2. Procjena zadovoljstva korištenjem proizvoda

- Opis cilja: Cilj procjene subjektivnog zadovoljstva je saznati koliko su korisnici zadovoljni korištenjem web portala. To se može procijeniti kroz njihovu opću reakciju, poput toga koliko im je portal bio jednostavan za korištenje ili koliko su se osjećali ugodno dok su ga koristili.
- Kriteriji uspjeha: Proizvod je uspješan ako korisnici izraze općenito pozitivno iskustvo bez većih pritužbi.

3. Evaluacija ispunjavanja kriterija uspjeha

- Opis cilja: Glavni cilj je usporediti rezultate testiranja s postavljenim ciljevima kako bi se vidjelo ispunjava li web portal očekivanja. Ako su rezultati dobri, proizvod je spreman; ako nisu, potrebna su poboljšanja.
- Kombiniranjem kvantitativnih podataka (npr. vrijeme, broj grešaka) i kvalitativnih uvida (npr. korisnički komentari) omogućava se sveobuhvatna ocjena korisničkog iskustva.

Za odabir metode mora se pružiti dovoljno informacija kako bi neovisni ispitivač mogao ponoviti postupak korišten u ispitivanju. Za testiranje korisničkog web portala prema CIF normi, testiranje upotrebljivosti je najprikladnija metoda. Ova metoda omogućuje direktno promatranje kako stvarni korisnici koriste web portal, pružajući vrijedne podatke o njihovoj izvedbi i zadovoljstvu. Međutim, zbog nemogućnosti dovoljno resursa istu nije bilo moguće koristiti stoga smo se odlučili za drugu metodu - heuristička evaluacija.

Stručnjaci za upotrebljivost (eng. usability experts) pregledavaju web portal prema skupu dobro poznatih heuristika (pravila), kao što su Nielsenove heuristike za korisničko iskustvo. Te smjernice obuhvaćaju aspekte poput jednostavnosti navigacije, konzistentnosti, povratnih informacija za korisnika, prevencije grešaka i drugih. Više stručnjaka obično sudjeluje u evaluaciji, svaki od njih identificira probleme i daje povratne informacije. Različiti evaluatori mogu otkriti različite probleme, što čini procjenu sveobuhvatnijom. Putem kvantitativne i kvalitativne analize stručnjaci ocjenjuju težinu svakog pronađenog problema, kategorizirajući ih prema ozbiljnosti (npr. manji, srednji ili kritični problemi). Zatim daju preporuke za poboljšanja.

Heuristička evaluacija je brža i manje skupa u usporedbi s testiranjem korisnika, jer ne zahtijeva angažman krajnjih korisnika i može se provesti ranije u procesu dizajna. Ova metoda je osobito korisna u ranim fazama razvoja, jer omogućava rano prepoznavanje

i ispravljanje problema u dizajnu. Prednosti heurističke evaluacije s CIF normom su efikasnost, stručna analiza i mogućnost rane primjene. Efikasnost postiže jer metoda omogućava brzo prepoznavanje problema u dizajnu, što može uštedjeti vrijeme i resurse. Stručna analiza se dobiva koristeći stručnjake za upotrebljivost i time se osigurava da se problemi procjenjuju prema dobro utemeljenim smjernicama. Mogućnost rane primjene vidljiva je jer se heuristička evaluacija može provesti već u ranim fazama razvoja, omogućujući da se problemi isprave prije nego što dođu do krajnjih korisnika.

Ukupan broj testiranih sudionika za razdoblje kroz 2022./2023.godinu je 5 ustanova. Stručnjaci su u prvoj fazi imitirali ustanove. Nakon testne implementacije kod korisnika svaka od tih ustanova zadatke je testirala individualno. Postoje dvije različite testne uloge, a to su: uloga testiranja kao običan korisnik i uloga testiranja kao administrator. Cilj segmentacije je osigurati da se ove dvije različite vrste korisnika evaluiraju na način koji je prilagođen njihovim potrebama, očekivanjima i ponašanjima. Segmentacija postoji zbog različite poslovne uloge.

- Ključne karakteristike i mogućnosti korisničke skupine.

Obični korisnik vidi samo svoje prijavljene zadatke, dok Administrator vidi prijavljene zadatke svih korisnika.

- Kako su odabrani sudionici; jesu li imali bitne karakteristike.

Kao što je navedeno, razlika između običnog korisnika i administratora je u poslovnoj ulozi. Obični korisnici su svi oni koji žele pratiti razvoj prijavljenog zadatka i otvoreni su prema novim tehnologijama. S takvim korisnicima je jednostavno surađivati jer se na brz način otkrije problem.

U ovoj situaciji stručnjaci koji prvi testiraju proizvod su već upoznati s takvim poslovnim ulogama stoga ima takav proces rada nije stran i znaju se postaviti u poziciju korisnika.

- Razlike između uzorka sudionika i korisničke populacije.

U prvom testiranju proizvoda dok još nije implementiran kod korisnika, stručni testeri testiraju na način na koji bi njihovi korisnici komunicirali s proizvodom i na taj način se eliminiraju prve greške (manje trošenje resursa). Nakon testne implementacije proizvoda, u suradnji s 5 korisnika testiran je proizvod. Na taj način smo također brže riješili greške (npr. problem u login-u) koje bi dobili kod ostalih korisnika. Time čekanje na ispravak postaje resursno i vremenski efikasnije.

U kontekstu upotrebe proizvoda tijekom testiranja, stručnjaci i korisnici su se ponašali na vrlo sličan način jer je iz dobivenih uputa bilo vrlo jasno kako se portal ponaša. U prvoj iteraciji testiranja korisnika, s 2 od njih 5 je bilo potrebno obaviti zajedničko provođenje testa. Kontekst upotrebe proizvoda u testu može se okarakterizirati kao simulacija realnog korištenja proizvoda.

Tijekom testiranja je važno da sudjeluju tester i različiti karakteristika. U vrednovanju korisničkog web portala je sudjelovalo 5 evaluatora. Svaki od njih je po prirodi različitog osobnog karaktera što pomaže u pogledu na test i samom njegovom izvođenju. Također, od njih pet, tri evaluatora su stručni tester i koji svaki od njih u svom dnevnom poslu ima dodijeljene različite domene testiranja. Ostala dva evaluatora su konzultanti koji su u čestoj komunikaciji s klijentima. Jedan tester i jedan konzultant imaju dodijeljenu domenu korisničkog portala i svakodnevno su se bavili analizom i testiranjem. Njihova mišljenja su jako važna zbog same upućenosti u domenu. Ali zbog većeg subjektivnog mišljenja, kategoriziranje prijavljenih grešaka po prioritetu se radilo u dogovoru s razvojem.

Tijekom internog testiranja portala prije nego je implementiran kod korisnika, tester i scenarije zadataka podijelili u nekoliko kategorija: Prijavljivanje zadataka, Pregled zadataka, Administracija korisnika, Dodavanje novog korisnika i Dodatne greške. Svaki od njih sadrži testne slučajeve koje je tester prošao. Testovi su kreirani na način da se tester postavi u poziciju korisnika i testira portal onako kako bi ga koristio korisnik. Zadaci su kategorizirani, stoga ako tester doda neki prijedlog tj. uviđa nedostatak, dodaje mu kategoriju novog zahtjeva. Zadaci takve kategorije se prijavljuju kroz klasičan način prijave te dodjeljuju dalje razvojnom timu.

– Zašto su odabrani ovi zadaci? (Najčešći zadaci, najproblematičniji zadaci..)

Ovi zadaci su odabrani jer se poznaje proces korisnika te su se testovi radili uz svu dokumentaciju i na temelju pretpostavke kako bi korisnik upotrebljavao proizvod.

– Scenariji zadataka i primjeri testova za testiranje:

1. Prijavljivanje zadataka:

„Unos prijave zadatka – naslov prijave, opis prijave, klik na prijavu i dodavanje više priloga“ ,

„Automatsko popunjavanje polja kontakta osobe koja prijavljuje zadatak“ ,

„Omogućiti drag&drop u dodavanju priloga“ ,

„Definiranje uloga unutar portala – ako je ulogirani korisnik u nekoj domeni, pri prijavi zadatka neka se automatski postavi označen proizvod domene i da nema pristup drugim domenama“.

2. Pregled zadataka:

„Dodavanje bilješke i priloga“,

„U pregledu zadataka prikazati defaultno sve zadatke koji nisu zatvoreni“,

„U pregledu zadataka postaviti defaultni poredak po datumu zadnje aktivnosti na zadataku“,

„Fiksirati pretragu po filterima da se ne miče kada se scroll-a na dno“,

„Optimizirati prikaz zadataka u razdoblju.“

3. Administracija korisnika:

„Nemogućnost brisanja korisnika“,

„Poruka ažuriranja korisnika se nastavlja javljati i nakon spremanja“,

„Ako promijenimo ulogu korisnika i ne pritisnemo [Spremi] nego [Aktiviraj/Deaktiviraj] - > uloga korisnika se vrati na onu koja je bila prije promjene“,

„U Administraciji korisnika promijenimo jednom korisniku status iz 'Administrator' u 'Običan korisnik' -> status se promijeni svim unesenim korisnicima u administraciji“,

4. Dodavanje novog korisnika:

„Dodati mogućnost vidljivosti proizvoda za korisnika“,

„Uspješno dodavanje korisnika s neispravnim podacima“,

„Moguće je više puta dodati korisnika s istim mailom“,

„Uvođenje prava na koje domene unutar projekta korisnik ima pravo pristupa, pregleda i notifikacija“.

5. Dodatne greške:

„Na portalu se ne vide svi zadaci iz Mantis“,

„Ispraviti pop-up prozor nakon aktivacije računa“,

„Cijeli portal postaviti da bude responsive“,

„Omogućiti da se prijave na mail dobivaju po određenoj domeni“.

Testiranje zadataka se vodi kroz tablicu koju tester mogu međusobno dijeliti u realnom vremenu (Google disk tablica). Razvojna strana također ima uvid u tablicu kako bi mogla lakše pratiti testne slučajeve i voditi evidenciju.

10.2 Rezultati testiranja web portala

Učinkovitost povezuje ciljeve korištenja proizvoda s točnošću i potpunosti kojom se ti ciljevi mogu postići. Uobičajene mjere učinkovitosti uključuju postotak dovršenosti zadatka, učestalost pogrešaka, učestalost pomoći sudioniku od strane ispitivača i učestalost pristupa pomoći ili dokumentaciji od strane sudionika tijekom zadataka. Ne uzima se u obzir način na koji su ciljevi zadatka postignuti, već samo stupanj do kojeg su postignuti. Učinkovitost povezuje razinu postignute učinkovitosti s količinom utrošenih resursa. Prema dobivenim podacima iz testiranja, svi korisnici koji su testirali su uspješno završili testiranje. Dobiveni podaci iz 5 kategorija prema testnim slučajevima iz tablice testera stručnjaka:

- Učinkovitost ispravljanja grešaka: metrika pokazuje koliko učinkovito tim ispravlja prijavljene greške prije isporuke.

Učinkovitost ispravljanja grešaka = $(\text{Zadaci s ispravljenim greškama prije isporuke} / \text{Ukupan broj testnih slučajeva}) \times 100$

- Stopa otkrivenih grešaka: metrika mjeri koliko se grešaka identificira u testnim slučajevima.

Stopa otkrivenih grešaka = $(\text{Zadaci s greškama prije isporuke} / \text{Ukupan broj testnih slučajeva}) \times 100$

- Stopa novih zahtjeva: metrika mjeri udio novih zahtjeva u ukupnom broju testnih slučajeva.

Stopa novih zahtjeva = $(\text{Novi zahtjevi} / \text{Ukupan broj testnih slučajeva}) \times 100$

- Stopa popravljenih grešaka: metrika mjeri koliko je grešaka uspješno ispravljeno u odnosu na prijavljene greške.

Stopa popravljenih grešaka = $(\text{Zadaci popravljenih grešaka} / \text{Zadaci s greškama prije isporuke}) \times 100$

- Stopa prihvaćenih nedostataka: metrika mjeri koliko je nedostataka prihvaćeno od strane razvojnog tima.

Stopa prihvaćenih nedostataka = (Ukupan broj prihvaćenih nedostataka / Ukupni prijavljeni nedostaci) × 100

- Omjer učinkovitosti testiranja: metrika mjeri učinkovitost testiranja u pronalaženju grešaka, pružajući uvid u to koliko su testni slučajevi bili uspješni u otkrivanju kritičnih problema.

Omjer učinkovitosti testiranja = (Zadaci s greškama / Ukupan broj testnih slučajeva) × 100

- Postotak uspješno riješenih grešaka: metrika prati koliko je uspješno riješenih grešaka u odnosu na sve prijavljene greške. Pomaže pri procjeni učinkovitosti tima u rješavanju otkrivenih problema.

Postotak uspješno riješenih grešaka = (Zadaci popravljenih grešaka / Ukupan broj prijavljenih grešaka) × 100

U nastavku se daje prikaz podataka o svakom setu zadataka koji su se koristili u testiranju web portala.

1. Prijavljivanje zadataka

Ukupan broj testnih slučajeva: 27

Zadaci s greškama koji su ispravljani prije isporuke: 9

Zadaci popravljenih grešaka: 8

Novi zahtjevi: 12

Učinkovitost ispravljanja grešaka: $(9 / 27) \times 100 = 33.33\%$

Stopa otkrivenih grešaka: $(9 / 27) \times 100 = 33.33\%$

Stopa novih zahtjeva: $(12 / 27) \times 100 = 44.44\%$

Stopa popravljenih grešaka: $(8 / 9) \times 100 = 88.89\%$

2. Pregled zadataka

Ukupan broj testnih slučajeva: 49

Zadaci s greškama koji su ispravljani prije isporuke: 16

Zadaci popravljenih grešaka: 11

Novi zahtjevi: 16

Učinkovitost ispravljanja grešaka: $(16 / 49) \times 100 = 32.65\%$

Stopa otkrivenih grešaka: $(16 / 49) \times 100 = 32.65\%$

Stopa novih zahtjeva: $(16 / 49) \times 100 = 32.65\%$

Stopa popraavljenih grešaka: $(11 / 16) \times 100 = 68.75\%$

3. Administracija korisnika

Ukupan broj testnih slučajeva: 12

Zadaci s greškama koji su ispravljani prije isporuke: 7

Zadaci popraavljenih grešaka: 4

Novi zahtjevi: 1

Učinkovitost ispravljanja grešaka: $(7 / 12) \times 100 = 58.33\%$

Stopa otkrivenih grešaka: $(7 / 12) \times 100 = 58.33\%$

Stopa novih zahtjeva: $(1 / 12) \times 100 = 8.33\%$

Stopa popraavljenih grešaka: $(4 / 7) \times 100 = 57.14\%$

4. Dodavanje novog korisnika

Ukupan broj testnih slučajeva: 7

Zadaci s greškama koji su ispravljani prije isporuke: 3

Zadaci popraavljenih grešaka: 2

Novi zahtjevi: 4

Učinkovitost ispravljanja grešaka: $(3 / 7) \times 100 = 42.86\%$

Stopa otkrivenih grešaka: $(3 / 7) \times 100 = 42.86\%$

Stopa novih zahtjeva: $(4 / 7) \times 100 = 57.14\%$

Stopa popraavljenih grešaka: $(2 / 3) \times 100 = 66.67\%$

5. Dodatne greške

Ukupan broj testnih slučajeva: 20

Zadaci s greškama koji su ispravljani prije isporuke: 6

Zadaci popraavljenih grešaka: 3

Novi zahtjevi: 11

Učinkovitost ispravljanja grešaka: $(6 / 20) \times 100 = 30\%$

Stopa otkrivenih grešaka: $(6 / 20) \times 100 = 30\%$

Stopa novih zahtjeva: $(11 / 20) \times 100 = 55\%$

Stopa popraavljenih grešaka: $(3 / 6) \times 100 = 50\%$

6. Stopa prihvaćenih nedostataka

Ukupan broj prihvaćenih nedostataka: 29

Ukupno prijavljeni nedostaci: 81

Stopa prihvaćenih nedostataka: $(29 / 81) \times 100 = 35.80\%$

7. Izračun učinkovitosti testiranja

Prijavljivanje zadataka: $(9 / 27) \times 100 = 33.33\%$

Pregled zadataka: $(16 / 49) \times 100 = 32.65\%$

Administracija korisnika: $(7 / 12) \times 100 = 58.33\%$

Dodavanje novog korisnika: $(3 / 7) \times 100 = 42.86\%$

Dodatne greške: $(6 / 20) \times 100 = 30\%$

8. Izračun uspješno riješenih grešaka

Prijavljivanje zadataka: $(8 / 81) \times 100 = 9.88\%$

Pregled zadataka: $(11 / 81) \times 100 = 13.58\%$

Administracija korisnika: $(4 / 81) \times 100 = 4.94\%$

Dodavanje novog korisnika: $(2 / 81) \times 100 = 2.47\%$

Dodatne greške: $(3 / 81) \times 100 = 3.70\%$

Cilj ovog istraživanja bio je procijeniti i validirati upotrebljivost web portala ERP sustava kroz različite metrike testiranja, a na temelju CIF (Common Industry Format) norme. Podaci prikupljeni kroz pet setova testnih slučajeva pružili su značajan uvid u područja koja zahtijevaju optimizaciju i poboljšanja. Omjer otkrivenih grešaka u testnim slučajevima varirao je između 30% i 58.33%, ovisno o specifičnim funkcionalnostima sustava. Posebno visoka učinkovitost testiranja u "Administraciji korisnika" (58.33%) ukazuje na dobar proces provjere u ovom dijelu sustava. Međutim, područja kao što su "Prijavljivanje zadataka" i "Dodatne greške" imaju niže stope otkrivanja, što upućuje na nedovoljno pokriveno testiranje. Prosječna stopa popravljenih grešaka bila je između 50% i 88.89%, što pokazuje visok stupanj učinkovitosti tima za razvoj u rješavanju identificiranih problema. Ovo je pozitivan znak jer ukazuje na brzu i učinkovitu reakciju tima u rješavanju problema. Visoka stopa novih zahtjeva u nekim dijelovima sustava, poput "Dodavanja novog korisnika" (57.14%) i "Dodatne greške" (55%), ukazuje na potrebu za boljem upravljanjem i dokumentiranjem zahtjeva korisnika. Moguće je da se neki novi zahtjevi javljaju zbog nedovoljne funkcionalnosti

ili potrebe za proširenjem značajki. Stopa prihvaćenih nedostataka od strane razvojnog tima je bila relativno niska (35.80%), što sugerira potrebu za boljom komunikacijom između testera i razvojnih timova ili preciznijom dokumentacijom grešaka.

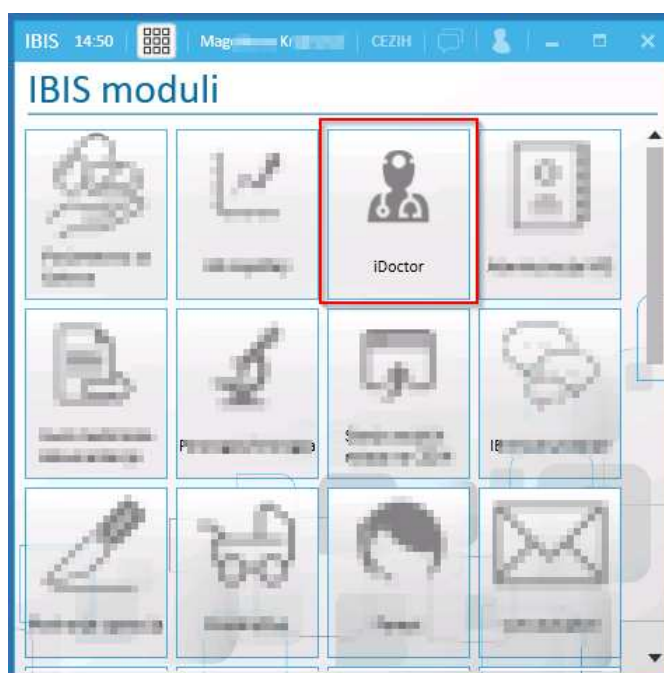
10.3 Validacija upotrebljivosti desktop i mobilne aplikacije

U ovom poglavlju daje se prikaz rezultata studije slučaja testiranja validacije upotrebljivosti aplikacija iDoctor i mVizita. Za potrebe validacije upotrebljivosti, obje aplikacije su testirane u stvarnim poslovnim okruženjima, pri čemu su se analizirali različiti aspekti korisničkog iskustva, uključujući vrijeme odgovora, funkcionalnost i intuitivnost sučelja na desktop i mobilnoj platformi. Unutar ERP sustava više različitih korisnika iz različitih poslovnih položaja koriste istu aplikaciju. Jedna se koristi kao desktop aplikacija, a druga kao mobilna. U nastavku empirijskog dijela rada ukratko je opisan rad aplikacije, vrednovanje i testiranje te naposljetku rezultati upotrebljivosti i optimizacije.

Proizvod koji je testiran u radu je aplikacija koju koristi medicinsko osoblje raznih zdravstvenih ustanova u republici HR. Aplikacija se koristi često i vrlo je važno da je funkcionalna te da se korisnik osjeća ugodno dok ju koristi. U aplikaciji koja je već neko vrijeme implementirana kod korisnika počinje imati određene probleme brzine rada. Ustanovljeno je usporenje rada proizvoda nakon što su isporučeni novi ispravci. Nakon analize i dijagnoze prijavljenih grešaka, budući da aplikaciju koriste korisnici različitih uloga, odlučeno je da je potrebno aplikaciju optimizirati. U testiranju prije isporuke ispravaka se mjerila brzina akcija modula po različitim korisnima. Na temelju dobivenih rezultata radili su se ispravci. Nakon ispravaka se regresijski testiralo sučelje na oba uređaja. Nakon isporuke ispravka, ponovno su se testirale iste akcije s istim korisnicima kako bi se dobili rezultati prema kojima se može usporediti uspješnost optimizacije koja utječe na upotrebljivost i korisničko iskustvo. Budući da se testiranje vršilo kod stvarnog korisnika, kako bi se zaštitili njegovi podaci, tijekom testiranja se isti zove Korisnik A1.

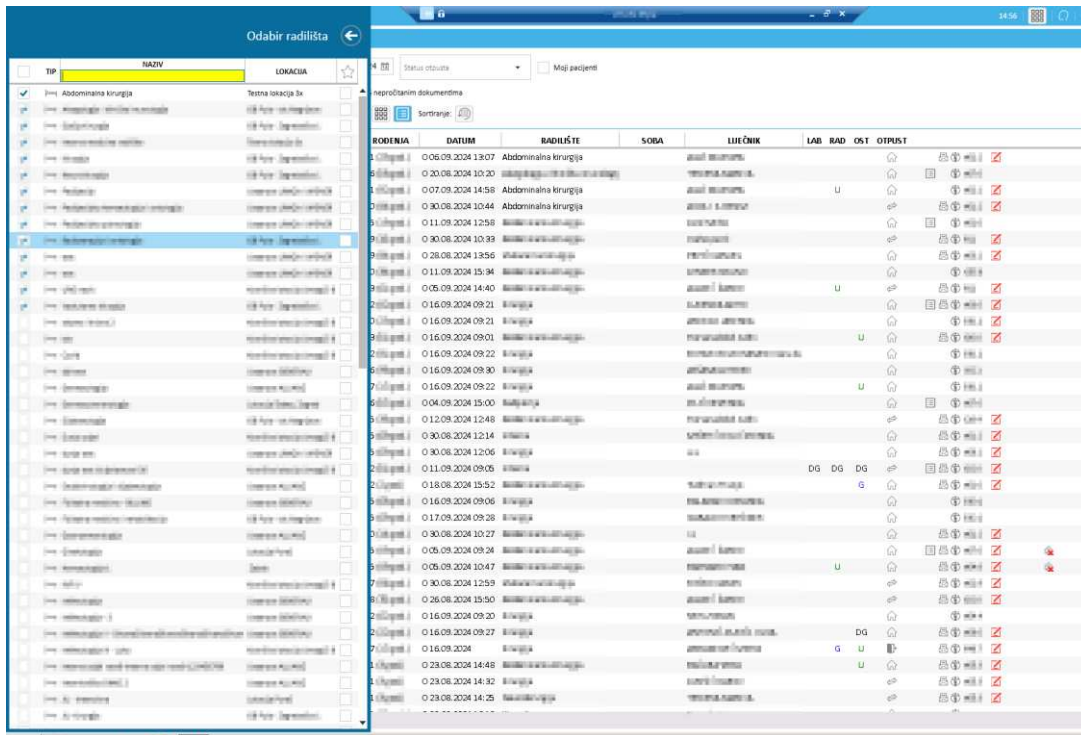
U nastavku se daje kratki opis desktop aplikacije koji se u testiranju imenovao Istraživanje 1:Korisnik A1. Putem aplikacije AppStarter, korisnici ulaze u modul iDoctor. Unutar modula moguće je odabrati željeno radilište na kojem se nalaze pacijenti, filtrirati ih i obrađivati neke podatke nad pacijentima. Odabirom pacijenta,

korisnik ima mogućnost pregleda tijeka događaja pacijenta, njegove arhive, te mu može dodavati ili micati neke dokumente. Modul je u koheziji s drugim vanjskim sustavima (razni laboratorijski sustavi i dr.) stoga se u procesu testiranja uzimaju svi realni scenariji kako bi modul radio ispravno. Dakle, nakon prijave u sustav, korisnik iz aplikacije AppStarter ulazi u modul iDoctor.



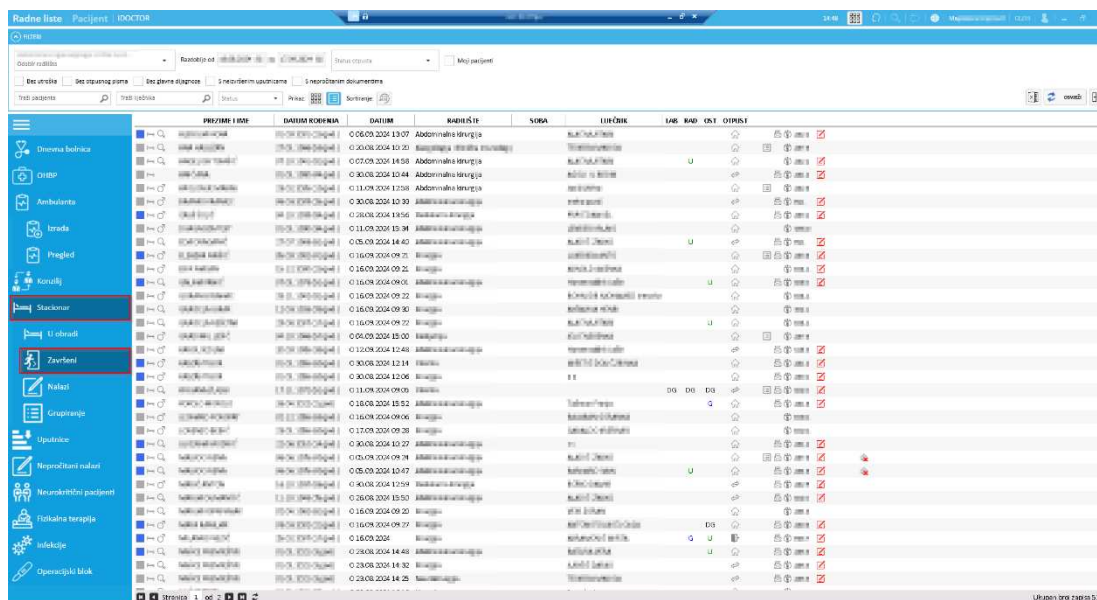
Slika 10 Prikaz aplikacije iDoctor, izvor: IBIS portal IN2 grupe

Nakon ulaska u modul, korisnik ima mogućnost odabrati radilišta i pozicionirati se na čvor željenog pacijenta. Prema primjeru slike, odabrano je radilište abdominalne kirurgije.



Slika 11 Odabir radilišta, izvor: IBIS portal IN2 grupe

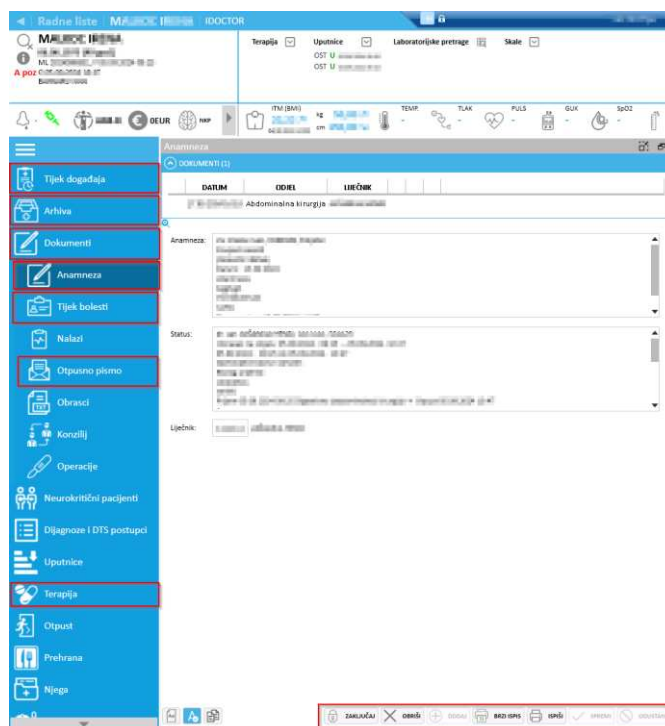
U nastavku korisnik odabire čvor na kojem se nalazi pacijent. Prema primjeru slike, korisnik je pozicioniran na čvoru Stacionara, podčvor 'Završeni' što znači da pacijent ima napisane dokumente koji su poslani na CEZIH i pacijent je završio svoje liječenje.



Slika 12 Odabir čvora unutar radne liste, izvor: IBIS portal IN2 grupe

Nakon što se korisnik pozicionira na željeni čvor i odabere pacijenta sa radne liste, otvara mu se pregled pacijenta kojem može vidjeti njegov tijek događaja, arhivu, pisati

mu neke dokumente (spremanje, zaključavanje, brisanje, ispis itd.), unositi terapiju, operaciju i drugo.



Slika 13 Pregled odabranog pacijenta iz radne liste, izvor: IBIS portal IN2 grupe

Akcije modula koje su korištene u testiranju su akcije koje se pamte po korisniku. Kroz ove akcije se mjerilo vrijeme početka akcije i završetka akcije. Neke od mjerenih akcija modula su: prvo i drugo otvaranje AppStarter-a, početak i završetak akcije klika ulaska u iDoctor, navigacija po čvorovima unutar iDoctora, prvo i drugo otvaranje pacijenta s radne liste, početak i završetak akcije klika na meni arhive, početak i završetak akcije klika otvaranja anamneze, početak i završetak akcije klika spremanja anamneze, početak i završetak akcije klika zaključavanja anamneze, početak i završetak akcije klika na meni tijekom događaja i drugo.

Budući da je prijavljeno usporeenje sustava Korisnika A1 , za početak su se uzele brzine akcija određenog korisnika unutar ustanove. U analizi su se testirale 52 akcije, što znači da su postojala 52 test slučaja. Izračunat je medijan svake akcije i prema tim rezultatima je uočeno koje akcije rade usporeno i treba ih optimizirati. Prema tim rezultatima, razvojni tim je radio optimizaciju. Nakon što je razvojni tim napravio optimizaciju, modul je regresijski testiran. Ispravak se isporučio kod korisnika te su se

ponovno testirane akcije modula. Nakon mjerenja, izračunao se medijan svake akcije te se razlikom dva medijana dobio postotak uspješnosti optimizacije.

U testiranju je sudjelovalo pet evaluatora. Od njih pet, tri evaluatora su stručni testeri koji svaki od njih u svom dnevnom poslu ima dodijeljene različite domene testiranja i zbog toga iz različitih perspektiva pristupaju svakom testu. Ostala dva evaluatora su konzultanti koji su inače u čestoj komunikaciji s klijentima, stoga su više upoznati u sam proces. Upravo zbog toga su se njihove evaluacije najviše uzimale u obzir.

10.3.1 Rezultati testiranja desktop aplikacije

Testiranje prikazuje da od 52 testna slučaja, njih 15 ima negativnu optimizaciju. Iz dobivenih vrijednosti su se izračunale metrike ukupnog prosječnog postotka poboljšanja, postotak uspješnih testnih slučajeva i postotak neuspjelih testnih poboljšanja/optimizacija.

Metrika ukupnog prosječnog postotka poboljšanja računa prosječni postotak poboljšanja za sve testne slučajeve i mjeri se prema formuli:

Prosječan postotak poboljšanja = $(\Sigma \text{POSTOTAK}) / (\text{Ukupan broj testnih slučajeva})$

Prosječan postotak poboljšanja za sve testne slučajeve iznosi 27,35%. To znači da je optimizacija poboljšala brzinu sustava u prosjeku za 27,35%.

Metrika postotka uspješnih testnih slučajeva se mjeri prema sljedećoj formuli:

Postotak uspješnih testnih slučajeva = $(\text{Broj testnih slučajeva s pozitivnim postotkom} / \text{Ukupan broj testnih slučajeva}) \times 100$

Izračunom je dobiveno da ukupni postotak testnih slučajeva s pozitivnim poboljšanjem iznosi 69,23%. Što znači da je skoro 70% optimizacija rezultiralo pozitivnim poboljšanjem performansi sustava.

Metrika postotak neuspješnih optimizacija se računa prema sljedećoj formuli:

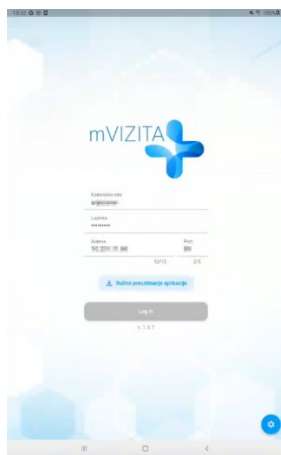
Postotak neuspjelih optimizacija = $(\text{Broj negativnih testnih slučajeva} / \text{Ukupan broj testnih slučajeva}) \times 100$

Iz formule se zaključuje da oko 28,85% optimizacija nije dalo očekivane rezultate i uzrokovalo je usporenje sustava.

Ove metrike ukazuju na značajno poboljšanje performansi, no postoji i prostor za daljnja poboljšanja u slučajevima gdje optimizacija nije bila učinkovita (oko 30% neuspješnih optimizacija)

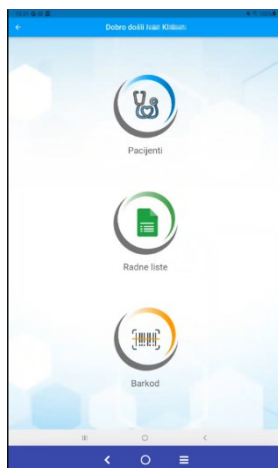
Na temelju provedenog testiranja u kojem je sudjelovalo pet evaluatora s različitim razinama stručnosti, dobiveni su rezultati koji pružaju sveobuhvatan uvid u optimizaciju ERP sustava. Tri stručna testera, svaki sa specifičnom domenom testiranja, su doprinijela različitim perspektivama u evaluaciji, dok su dva konzultanta, zbog česte komunikacije s krajnjim korisnicima, imala ključnu ulogu u procjeni jer bolje razumiju stvarne poslovne procese korisnika. Rezultati pokazuju da je od ukupno 52 testna slučaja, njih 15 (28,85%) imalo negativnu optimizaciju, što ukazuje na određena usporenja sustava. S druge strane, 69,23% testnih slučajeva pokazalo je pozitivna poboljšanja u performansama, što ukazuje na uspjeh većine optimizacija. Prosječni postotak poboljšanja za sve testne slučajeve iznosi 27,35%, što znači da je optimizacija u prosjeku povećala brzinu sustava za više od četvrtine. Prisutnost od gotovo 30% neuspješnih optimizacija sugerira potrebu za daljnjom analizom i poboljšanjima. Treba detaljnije ispitati razloge zbog kojih pojedine optimizacije nisu postigle željene rezultate kako bi se minimizirali rizici budućih usporenja i osigurala veća dosljednost u poboljšanju performansi sustava. Sveukupno, testiranje je pokazalo da optimizacije ERP sustava donose opipljive koristi, ali je neophodno uložiti dodatne napore kako bi se smanjila stopa neuspjeha u preostalim slučajevima.

U nastavku se daje kratki opis mobilne aplikacije koji se u testiranju imenovao Istraživanje 2:Korisnik A2. Drugo istraživanje se vršilo na isti način na istoj aplikaciji koju koriste zdravstveni djelatnici, samo što se aplikacija koristila na mobilnom uređaju. Aplikaciju također koriste djelatnici raznih poslovnih funkcija. Dizajnom aplikacija djeluje drugačije, ali je u suštini jednaka.



Slika 14 Login u aplikaciju mVizita, izvor: mVizita portal IN2 grupe

Nakon što se korisnik ulogira u sustav, ima mogućnost odabrati tri opcije za nastavak rada, a one su Pacijenti, Radne liste i Barkod. Odabirom Pacijenti korisnik može pregledati sve pacijente, odabirom Radne liste korisnik ima pregled svih aktivnosti toga dana. Odabirom Barkod korisnik izrađuje barkod naljepnice pacijentima na duljem boravku.



Slika 15 Login u aplikaciju mVizita, izvor: mVizita portal IN2 grupe

Kada korisnik odabere nekog pacijenta, ima mogućnosti odrađivati procese kao i korisnik unutar desktop aplikacije iDoctor.



Slika 16 Pregled podataka o pacijentu, izvor: mVizita portal IN2 grupe

U testiranju su sudjelovala tri evaluatora. Od toga, dva evaluatora su stručnjaci koji u svakodnevnom radu provode testiranja na različitim domenama što dovodi do više perspektiva u pristupu testiranju. Također, postojala je i ograničenost uređaja, stoga veća skupina ljudi nije mogla testirati. Zadnji evaluator je konzultant koji je inače u

često komunikaciji s klijentima, stoga je više upoznat u sam proces. Zbog toga su se njegove evaluacije najviše uzimale u obzir.

10.3.2 Rezultati testiranja mobilne aplikacije

Testiranje prikazuje da od 29 testna slučaja, njih 10 ima negativnu optimizaciju. Ostalih 19 je dalo pozitivnu optimizaciju. Iz dobivenih vrijednosti su se izračunale metrike ukupnog prosječnog postotka poboljšanja, postotak uspješnih testnih slučajeva i postotak neuspjelih testnih poboljšanja/optimizacija.

Metrika ukupnog prosječnog postotka poboljšanja računa prosječni postotak poboljšanja za sve testne slučajeve i mjeri se prema formuli:

Prosječan postotak poboljšanja = $(\Sigma \text{POSTOTAK}) / (\text{Ukupan broj testnih slučajeva})$

Prosječan postotak poboljšanja za sve testne slučajeve iznosi 22,69%. To znači da je optimizacija poboljšala brzinu sustava u prosjeku za 22,69%.

Metrika postotka uspješnih testnih slučajeva se mjeri prema sljedećoj formuli:

Postotak uspješnih testnih slučajeva = $(\text{Broj testnih slučajeva s pozitivnim postotkom} / \text{Ukupan broj testnih slučajeva}) \times 100$

Izračunom je dobiveno da ukupni postotak testnih slučajeva s pozitivnim poboljšanjem iznosi 65,52%. Što znači da je pregled uspješnosti optimizacije primjenjiv na sustav nakon isporuke ispravka.

Metrika postotak neuspješnih optimizacija se računa prema sljedećoj formuli:

Postotak neuspjelih optimizacija = $(\text{Broj negativnih testnih slučajeva} / \text{Ukupan broj testnih slučajeva}) \times 100$

Iz formule se zaključuje da oko 34,48% optimizacija nije dalo očekivane rezultate i uzrokovalo je usporenje sustava. Što je više u odnosu na primjer korisnika A1.

Na temelju provedenog testiranja dobiveni su rezultati koji pružaju zanimljiv uvid u optimizaciju ERP sustava. Dva evaluatora s bogatim iskustvom u testiranju različitih domena su doprinijela široj perspektivi prilikom evaluacije. Treći evaluator je konzultant s čestim kontaktima s klijentima, imao je ključnu ulogu u testiranju zbog boljeg razumijevanja stvarnih potreba korisnika. Od ukupno 29 testnih slučajeva, njih 19 (65,52%) pokazalo je pozitivne rezultate optimizacije, što ukazuje na značajno poboljšanje performansi sustava nakon implementacije ispravaka. Prosječan postotak poboljšanja iznosio je 22,69%, što znači da je optimizacija ubrzala rad sustava u

prosijeku za gotovo četvrtinu. Ovaj rezultat potvrđuje da su poduzete mjere bile uglavnom uspješne u poboljšanju korisničkog iskustva. Međutim, 10 testnih slučajeva (34,48%) pokazalo je negativnu optimizaciju, što ukazuje na određena usporenja u radu sustava. Ova razina neuspjelih optimizacija upućuje na dodatnu analizu i korigiranje specifičnih segmenata sustava koji uzrokuju loše rezultate. U usporedbi s istraživanjem desktop verzije, postotak neuspjelih optimizacija na mobilnoj verziji je veći. Time se pokazuje veća potreba za dodatnim poboljšanjima u sljedećim fazama razvoja. Iako je većina optimizacija rezultirala poboljšanjem performansi i dalje postoji prostor za unapređenje kako bi se smanjio postotak neuspjelih testova i osiguralo stabilno i dosljedno poboljšanje u svim segmentima ERP proizvoda.

Dobiveni rezultati oba istraživanja se mogu usporediti. Testiranje desktop aplikacije je provelo pet evaluatora, dok je testiranje mobilne aplikacije provelo troje evaluatora. U oba slučaja uključeni su stručni testeri i konzultanti. Kroz istraživanje se zaključuje kako je uloga konzultanata u testiranju važnija od uloge stručnih testera. Navedeno proizlazi iz činjenice što konzultanti imaju kontinuirani dijalog s korisnicima i prvi su koji se susreću s njihovim povratnim informacijama upotrebljivosti, stoga dobro poznaju procesne korake i reakcije.

U istraživanju desktop aplikacije testirana su 52 testna slučaja, od kojih je 37 imalo pozitivnu optimizaciju. U istraživanju mobilne aplikacije testirano je 29 testnih slučajeva, od kojih je 19 imalo pozitivnu optimizaciju. Sličan je postotak pozitivnih rezultata u oba istraživanja, za desktop aplikaciju iznosi 69,23%, a za mobilnu aplikaciju iznosi 65,52%. Prosječni postotak poboljšanja za skoro 5% je veći u desktop aplikaciji naspram mobilne aplikacije čiji postotak iznosi 22,69%. Suprotno tome, negativne optimizacije za skoro 5% su veće kod mobilne aplikacije naspram desktop aplikacije čiji postotak iznosi 28,85%.

U oba istraživanja uočava se potreba za daljnjom analizom i poboljšanjem kako bi se smanjila stopa neuspješnih optimizacija i osigurala veća dosljednost u performansama ERP sustava, međutim u slučaju desktop aplikacije je veća potreba za kontinuiranim unaprjeđenjem jer gotovo 30% slučajeva nije postiglo željene rezultate. Kvantitativne rezultate istraživanja potkrepljuju povratne informacije korisnika prikupljene u komunikaciji s konzultantima. Oba istraživanja naglašavaju važnost optimizacije ERP sustava i prepoznaju prostor za daljnje poboljšanje performansi. Prvo istraživanje daje nešto širi i detaljniji pregled, zbog većeg broja evaluatora i testnih slučajeva, s

naglaskom na potrebu za dodatnom analizom i unapređenjem kako bi se smanjila stopa negativnih optimizacija.

Zaključak

Upotrebljivost ERP sustava ključna je za uspješnost implementacije i dugoročno funkcioniranje unutar organizacije. ERP sustavi kao složeni softverski proizvodi integriraju različite poslovne procese, uključujući nabavu, računovodstvo, prodaju i upravljanje zalihama, te omogućuju centraliziranu obradu podataka. Visoka razina upotrebljivosti doprinosi povećanoj učinkovitosti, smanjenju operativnih pogrešaka i boljem korisničkom zadovoljstvu, što izravno utječe na produktivnost poduzeća. Korisnici ERP sustava dolaze iz različitih dijelova organizacije te je ključno da su sustavi intuitivni i lako razumljivi, bez obzira na tehničku razinu znanja korisnika. Visoka upotrebljivost omogućuje jednostavniju obuku novih korisnika i smanjuje vrijeme potrebno za prilagodbu sustavu. Nadalje, upotrebljiv ERP sustav pomaže organizacijama da brže odgovore na promjene u poslovanju te lakše implementiraju nove procese i strategije. Optimizacija korisničkog sučelja i jednostavnost korištenja poboljšavaju operativnu učinkovitost te smanjuju troškove održavanja i podrške sustava. Kvalitetno testiranje upotrebljivosti, koje uključuje heurističke evaluacije i povratne informacije korisnika, omogućuje pravovremeno otkrivanje problema te prilagodbu sustava kako bi se osigurala dugoročna održivost i zadovoljstvo korisnika. Prilikom ručnog testiranja ERP proizvoda, ključno je pratiti korisničke zahtjeve i uzimati u obzir njihovo mišljenje kako bi se osiguralo da konačni proizvod u potpunosti zadovoljava potrebe korisnika. Korisnički zahtjevi definiraju funkcionalnosti i očekivanja koje web proizvod mora ispuniti, a ručno testiranje omogućuje da se te funkcionalnosti detaljno provjere iz perspektive stvarnog korisnika. Sve ove komponente čine upotrebljivost ERP sustava jednim od ključnih faktora uspjeha u dinamičnom poslovnom okruženju.

Provedena validacija upotrebljivosti ERP sustava na web portalu, desktop i mobilnoj aplikaciji pružila je zanimljiv uvid u performanse i korisničko iskustvo kroz različite platforme. Iako su rezultati općenito pokazali poboljšanja u funkcionalnosti i brzini rada, testiranja su također otkrila određena ograničenja i potencijalne probleme koje je potrebno dodatno analizirati i riješiti. Jedno od glavnih ograničenja provedenog testiranja bilo je vrijeme i opseg u kojem su korisnici mogli testirati sve funkcionalnosti sustava. Iako su testovi bili dizajnirani tako da pokriju ključne poslovne procese,

stvarna upotreba sustava u svakodnevnim uvjetima mogla bi otkriti dodatne probleme koje simulirani testovi nisu obuhvatili. Također, uzorak korisnika koji su sudjelovali u testiranju je bio relativno ograničen na određene poslovne uloge. Osim toga, testiranje je bilo fokusirano na mjerljive aspekte performansi, poput brzine odziva i optimizacije sustava, a kvalitativni aspekti korisničkog iskustva, poput subjektivnog osjećaja zadovoljstva korisnika, zahtijevali bi dodatne upitnike ili intervjue za dublju analizu.

Kako bi se poboljšala valjanost budućih testova, predlaže se povećanje uzorka korisnika koji sudjeluju u testiranju, uključujući one iz različitih poslovnih područja i s različitim razinama tehničke stručnosti. Širi uzorak može otkriti specifične probleme koji nisu bili evidentni u inicijalnoj fazi testiranja, a koji mogu značajno utjecati na svakodnevnu upotrebu sustava. Također, preporučuje se uvođenje dugotrajnijeg testiranja u stvarnim poslovnim uvjetima, koje bi obuhvatilo korištenje aplikacija u različitim vremenskim razdobljima i pod različitim opterećenjima. To bi omogućilo identificiranje mogućih performansnih uskih grla, kao i dugoročnog utjecaja optimizacija na stabilnost i učinkovitost sustava. Specifično za mobilnu aplikaciju, predlaže se testiranje na različitim uređajima i operativnim sustavima kako bi se osiguralo da optimizacije i funkcionalnosti sustava rade jednako učinkovito bez obzira na platformu ili hardversku konfiguraciju.

Jedan od ključnih izazova u testiranju ERP sustava, općenito, jest balansiranje između složenosti poslovnih procesa koje sustav podržava i jednostavnosti korisničkog iskustva. Stoga je nužno da se testiranja ne fokusiraju samo na tehničke aspekte, već i na korisničku perspektivu kako bi sustav bio intuitivan i efikasan za sve korisnike. U kontekstu provedenih studija slučaja, rezultati su pokazali da je optimizacija sustava dovela do poboljšanja performansi u većini testiranih slučajeva, no postoji značajan postotak akcija koje su nakon optimizacije zabilježile negativan učinak. To ukazuje na potrebu za iterativnim pristupom optimizaciji, gdje bi se svaka promjena temeljito testirala prije uvođenja u širu upotrebu.

Literatura

1. Bastardo, R., Pavão, J., i Rocha, N.P. Methodological Quality of User-Centered Usability Evaluation of Digital Applications to Promote Citizens' Engagement and Participation in Public Governance: A Systematic Literature Review.
2. Bradford, M. (2015). Modern ERP: Select, Implement, & Use Today's Advanced Business Systems (3rd ed.).
3. PeopleCert International Ltd. (2023). Certified Associate of Software Testing (CAST). Cyprus: PeopleCert International Ltd.
4. Hamilton, T. (2024). What is software testing life cycle (STLC)?
5. Hvannberg, E.T., Law, E.L.C., i Lérusdóttir, M.K. (2007). Heuristic evaluation: Comparing ways of finding and reporting usability problems. *Interacting with Computers*.
6. Interaction Design Foundation (IxDF). (2016, May 25). What is heuristic evaluation (HE)?
7. Interaction Design Foundation (IxDF). (2016, November 20). How to conduct a heuristic evaluation for usability in HCI and information visualization.
8. ISO/IEC 25062:2006, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability test reports.
9. Kaizo. (2023). How to measure customer satisfaction.
10. Kan, S. H. (2002). *Metrics and models in software quality engineering*. Addison-Wesley.
11. Kaner, C., i Bach, J. (2001). *Lessons learned in software testing*. Wiley.
12. Katalon. (2024). *Functional testing: Ensuring your software works as expected*.
13. Lewis, J.R., i Sauro, J. (2021). *Usability and user experience: Design and evaluation*.
14. Matende, S., i Ogao, P. (2013). *Enterprise resource planning (ERP) system implementation: A case for user participation*.
15. Myers, G.J., Sandler, C., i Badgett, T. (2013). *The art of software testing* (3rd ed.). Wiley.
16. Nielsen, J. (1994a). *Usability inspection methods*. Wiley & Sons.

17. Nielsen, J. (1994b). Usability engineering. Morgan Kaufmann.
18. Ohno, T. (1988). Toyota production system: Beyond large-scale production. Productivity Press.
19. Plantak Vukovac, D., i Orehovački, T. (2010). Primjena heurističke evaluacije u analizi upotrebljivosti akademskih web sustava. In Proceedings of the MIPRO 2010 Conference on Computers in Education. Opatija, Croatia.
20. Sommerville, I. (2016). Software engineering (10th ed.). Pearson.
21. Sommerville, I. (2020). Engineering software products: An introduction to modern software engineering. Pearson.
22. Testsigma. (2024a). Manual testing: A complete guide to software testing.
23. Testsigma. (2024b). Software testing standards.
24. Wong, E. (2024, February 21). Heuristic evaluation: How to conduct a heuristic evaluation.

Sažetak

Ovaj diplomski rad se bavi validacijom upotrebljivosti ERP sustava, uključujući web portal, desktop i mobilnu aplikaciju. Glavni cilj bio je testirati i ocijeniti performanse ovih aplikacija kroz različite aspekte korisničkog iskustva, uključujući učinkovitost, jednostavnost korištenja i optimizaciju nakon uvođenja poboljšanja. Korištenje ISO/IEC 25062:2006 (CIF) norme osiguralo je strukturirani pristup evaluaciji sustava. Diplomski rad detaljno opisuje metode testiranja, uključujući heurističku evaluaciju, te naglašava važnost kontinuiranog testiranja za postizanje visokokvalitetnog softvera. Rezultati testiranja pokazali su poboljšanja u korisničkom iskustvu nakon optimizacije, ali su također otkrili neka usporenja u određenim procesima, što ukazuje na potrebu za dodatnim iterativnim poboljšanjima.

Ključne riječi: ERP sustavi, upotrebljivost, validacija, korisničko iskustvo, heuristička evaluacija, ISO/IEC 25062, optimizacija

Abstract

This master's thesis focuses on the usability validation of ERP systems, including the web portal, desktop, and mobile applications. The main goal was to test and evaluate the performance of these applications through various aspects of user experience, such as efficiency, ease of use, and optimization after improvements. The use of the ISO/IEC 25062:2006 (CIF) standard ensured a structured approach to system evaluation. The master's thesis describes the testing methods in detail, including heuristic evaluation, and highlights the importance of continuous testing to achieve high-quality software. Testing results showed improvements in user experience after optimization, but also revealed some slowdowns in certain processes, indicating the need for further iterative improvements.

Keywords: ERP systems, usability, validation, user experience, heuristic evaluation, ISO/IEC 25062, optimization

Popis slika

Slika 1 Razlika u kvaliteti softvera između proizvođača i korisnika	13
Slika 2 Aktivacija računa.....	36
Slika 3 Aktivan račun.....	37
Slika 4 Login sučelje.....	37
Slika 5 Obavijest o prijavi zadatka	38
Slika 6 Pregled zadataka.....	38
Slika 7 Običan korisnik i Administrator	39
Slika 8 Prikaz statusa zadatka.....	39
Slika 9 Prikaz statusa zatvorenog zadatka	40
Slika 10 Prikaz aplikacije iDoctor	53
Slika 11 Odabir radilišta	54
Slika 12 Odabir čvora unutar radne liste.....	54
Slika 13 Pregled odabranog pacijenta iz radne liste	55
Slika 14 Login u aplikaciju mVizita.....	57
Slika 15 Login u aplikaciju mVizita.....	58
Slika 16 Pregled podataka o pacijentu.....	58

Popis tablica

Tablica 1 Townsend-ov pogled na kvalitetu	13
Tablica 2 Lista normi.....	23