TaF- The healthy Chef

Plan testiranja

Verzija 1.1

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 14.05.2022. | 1.0 | Inicijalna verzija | Dušan |
| 15.05.2022. | 1.1 | Dorada dokumenta | Dušan |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Ciljevi 4

2. Opseg 4

3. Reference 4

4. Zahtevi za testiranje 4

5. Strategije testiranja (Test Strategy) 7

5.1 Tipovi testiranja (Testing Types) 7

5.1.1 Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing) 7

5.1.2 Testiranje sistema (System Testing) 7

5.1.3 Testiranje poslovnog ciklusa (Business Cycle Testing) 8

5.1.4 Testiranje korisničkog interfejsa (User Interface Testing) 9

5.1.5 Testiranje performansi (Performance Testing) 9

5.1.6 Testiranje opterećenja (Load Testing) 10

5.1.7 Testiranje kritičnih slučajeva(Stress Testing) 10

5.1.8 Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing) 11

5.1.9 Testiranje konfiguracije (Configuration Testing) 12

5.2 Alati 12

6. Resursi 13

6.1 Radnici 13

6.2 Sistem 14

7. Kontrolne tačke testiranja 15

8. Izlazni produkti testiranja 15

8.1 Model testiranja 15

9. Projektni zadaci 16

Plan Testiranja

# Ciljevi

Ovaj dokument sadrži opis plana testiranja aplikacije TaF. Namena ovog dokumenta je da opiše način provere funkcionalnih i nefunkcionalnih zahteva, postavljenih pri realizaciji predložene aplikacije. Ciljevi ovog dokumenta su:

* Izdvojiti komponente sistema koje treba testirati na osnovu postojećih informacija o projektu.
* Izdvojiti zahteve testiranja na osnovu datih predloga (na višem nivou).
* Predložiti i opisati strategije testiranja koje će biti korišćene.
* Odrediti resurse koji će biti potrebni za testiranje i proceniti vreme i trud koji treba uložiti pri testiranju.
* Navesti dokumente koji će opisati rezultate testiranja.

# Opseg

Plan testiranja se odnosi na integraciju aplikacije i proveru izvršne verzije aplikacije TaF.

Plan testiranja ne opisuje testiranje pojedinačnih komponenti sistema, jer se pretpostavlja da je svaki od programera izvršio kompletno testiranje koda pre njegovog dodavanja u sistem, korišćenjem metode crne kutije.

Plan testiranja će se odnositi na:

* testiranje interfejsa prema različitim tipovima korisnika,
* testiranje komunikacije sa bazom podataka,
* testiranje komunikacije komponenti aplikacije,
* testiranje funkcionalnosti izvršne verzije aplikacije i
* testiranje kritičnih performansi aplikacije.

# Reference

Reference na osnovu kojih će biti napisan Plan testiranja su:

1. Vizija sistema TaF,
2. Specifikacija zahteva aplikacije TaF i
3. Detaljno – arhitekturni projekat TaF.

# Zahtevi za testiranje

U ovom odeljku biće navedeni oni elementi aplikacije koje je potrebno testirati (slučajevi korišćenja, funkcionalni i nefunkcionalni zahtevi). Ovde će biti opisano samo šta je potrebno testirati, dok će detalji o samim testovima biti navedeni kasnije u dokumentu Test specifikacija.

**Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing)**

Proveriti mogućnost pristupa bazi TaF.

Proveriti mogućnost istovremenog pristupa istim podacima radi čitanja.

Proveriti tačnost podataka nakon ažuriranja istih.

Proveriti tačnost podataka koji se pribavljaju iz baze.

Proveriti da li izbrisani podaci i dalje postoje.

**Testiranje sistema – funkcionalno testiranje (System Testing – functional testing)**

Proveriti slučaj korišćenja Čitanje blogova.

Proveriti slučaj korišćenja Čitanje recepta.

Proveriti slučaj korišćenja Kreiranje naloga.

Proveriti slučaj korišćenja Prijavljivanje(Login).

Proveriti slučaj korišćenja Pregled preporučenih blogova za čitanje.

Proveriti slučaj korišćenja Pregled preporučenih recepta za čitanje.

Proveriti slučaj korišćenja Sortiranje recepta po datumu objavljivanja.

Proveriti slučaj korišćenja Soriranje recepata po vremenu spremanja obroka.

Proveriti slučaj korišćenja Sortiranje recepata po popularnosti.

Proveriti slučaj korišćenja Soritranje blogova po datumu objavljivanja.

Proveriti slučaj korišćenja Sortiranje blogova po popularnosti.

Proveriti slučaj korišćenja Dodela komentara receptu.

Proveriti slučaj korišćenja Dodela komentara blogu.

Proveriti slučaj korišćenja Dodela ocene receptu.

Proveriti slučaj korišćenja Dodela ocene blogu.

Proveriti slučaj korišćenja Ažuriranje podataka o sebi.

Proveriti slučaj korišćenja Uvid profila nekog autora i njegovog sadržaja.

Proveriti slučaj korišćenja Čekiranje izvršenih koraka nekog recepta.

Proveriti slučaj korišćenja Čuvanje omiljenih recepta.

Proveriti slučaj korišćenja Čuvanje omiljenih blogova.

Proveriti slučaj korišćenja Pregled sačuvanih recepata.

Proveriti slučaj korišćenja Pregled sačuvanih blogova .

Proveriti slučaj korišćenja Kreiranje bloga.

Proveriti slučaj korišćenja Kreiranje recepata.

Proveriti slučaj korišćenja Izmena sadržaja recepata.

Proveriti slučaj korišćenja Izmena sadržaja u blogu.

Proveriti slučaj korišćenja Brisanje recepata.

Proveriti slučaj korišćenja Brisanje bloga.

**Testiranje poslovnog ciklusa (Business Cycle Testing)**

Proveriti posledice operacije kreiranje naloga autora i čitaoca.

Proveriti posledice operacije kreiranje bloga i recepta.

Proveriti posledice operacije dodele komentara blogu i receptu.

Proveriti posledice operacije dodele ocene blogu i receptu.

Proveriti posledice operacije dodavanja bloga i recepta za čitanje kasnije.

**Testiranje korisničkog interfejsa (User Interface Testing)**

Proveriti jednostavnost navigacije kroz skup prozora aplikacije.

Proveriti da li izgled prozora aplikacije odgovara definisanim standardima za izradu korisničkog interfejsa.

Proveriti da li korisnički interfejs omogućava jednostavno i intuitivno korišćenje bez potrebe za organizovanjem dodatne obuke (Specifikacija zahteva, odeljak 7.2).

Aplikacija treba da bude laka za korišćenje i prilagođena korisnicima koji poseduju dobar nivo znanja rada na računaru (Vizija sistema, odeljak 5.2).

Proveriti da li *online* uputstvo obezbeđuje adekvatnu podršku pri korišćenju naprednijih funkcionalnosti sistema (Vizija sistema, odeljak 12.2).

**Testiranje performansi (Performance Testing)**

Proveriti vreme potrebno za prijavljivanje na aplikaciju.

Proveriti vreme potrebno za kreiranje naloga.

Proveriti vreme potrebno za sortiranje sadržaja(blogova ili recepata).

Proveriti vreme potrebno za kreiranje bloga i recepata.

Proveriti vreme potrebno za kreiranje komentara.

Proveriti vreme potrebno za dodavanje ocene blogu i receptu.

Proveriti da li vreme potrebno za pristupanje bazi podataka u cilju izvršenje upita ne prelazi 5 sekundi (Specifikacija zahteva, odeljak 7.4).

**Testiranje opterećenja (Load Testing)**

Proveriti odziv aplikacije kada mu istovremeno pristupa 200 posetilaca.

Proveriti odziv aplikacije kada 100 posetilaca istovremeno pristupa stranici Pregled recepata.

Proveriti odziv aplikacije kada 100 posetilaca istovremeno pristupa stranici Pregled blogova.

Proveriti da li sistem može da podrži do 1000 simultanih pristupa korisnika portalu (Detaljni arhitekturni projekat, odeljak 11).

**Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing)**

Proveriti mogućnost prijavljivanja na sistem sa računara u lokalnoj mreži.

Proveriti mogućnost prijavljivanja na sistem sa udaljenog računara.

Proveriti prava pristupa korisnika koji pripadaju različitim grupama (Posetilac portala, Autor, Čitaoc).

**Testiranje otkaza i oporavka (Failover / Recovery Testing)**

Proveriti da li je TaF aplikacija dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupan ne sme da pređe 10%. (Vizija sistema, odeljak 9; Detaljni arhitekturni projekat, odeljak 12)

Proveriti da li srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati (Detaljni arhitekturni projekat, odeljak 12).

**Testiranje konfiguracije (Configuration Testing)**

Severska komponenta sistema treba da funkcioniše na računaru pod operativnim sistemom Windows.

Klijentska komponenta sistema treba da funkcioniše na bilo kom PC računaru koji podržava savremene pretraživače. (Specifikacija zahteva, odeljak 7.6).

Proveriti da li klijentski deo TaF portala može da se koristi pomoću Web čitača: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, Edge, Chrome kao i Firefox (Mozilla) (Specifikacija sistema, odeljak 7.6).

# Strategije testiranja (Test Strategy)

Strategije testiranja predstavljaju preporuke na koji način treba vršiti testiranje softverske aplikacije. Prethodni odeljak o Zahtevima testiranja opisuje šta će biti testirano, dok će u ovom odeljku biti opisano kako će navedeni elementi biti testirani.

Osnova svake strategije testiranja jesu tehnike koje se koriste i kriterijumi na osnovu kojih se može utvrditi kada je testiranje završeno.

## Tipovi testiranja (Testing Types)

### Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing)

|  |  |
| --- | --- |
| Cilj Testa: | Utvrditi da metodi i procesi za pristup bazi podataka funkcionišu korektno i bez grešaka po podacima. |
| Tehnike: | 1. Probati svaki metod pristupa bazi podataka, za korektne i nekorektne podatke (ili za pribavljanje podataka) 2. Ispitati bazu podataka da bi se utvrdilo da se podaci unose kao što se očekuje, da svi događaji u bazi su obrađuju korektno i da se podaci pribavljaju korektno i bez grešaka. |
| Kriterijum završetka: | Metodi i procesi za pristup bazama podataka rade kao što je predviđeno i bez grešaka u podacima. |
| Napomene: | 1. Testiranje zahteva korišćenje okruženja za rad sa konkretnim DBMS-om da bi se direktno uneli i promenili podaci u bazi podataka. 2. Procesi moraju da se ručno pokrenu. 3. Koristiti bazu sa malim ili minimalnim brojem podataka, kako bi se povećala preglednost i lakše uočili neprihvatljivi slučajevi. |

### Testiranje sistema (System Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi korektno kretanje kroz aplikaciju, unos podataka, njihovu obradu i pribavljanje. |
| Tehnike: | 1. Izvršiti svaki slučaj korišćenja, ispratiti sve tokove definisane slučajevima korišćenja, korišćenjem tačnih i netačnih podataka. 2. Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima. 3. Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. 4. Potvrditi da su poslovna pravila pravilno primenjena. |
| Kriterijum završetka: | 1. Svi planirani testovi su izvršeni. 2. Sve identifikovane greške su adekvatno obrađene. |
| Napomene: | 1. Potrebno je obezbediti pristup serveru na kome će biti instaliran sistem. |

### Testiranje poslovnog ciklusa (Business Cycle Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi odgovarajuću primenu i procese koji se obavljaju kao podrška zahtevanom modelu poslovanja i utvrđenom rasporedu. |
| Tehnike: | 1. Testiranje simulira poslovne cikluse koji obavljaju sledeće: 2. Testovi korišćeni za testiranje funkcija aplikacije treba modifikovati/proširiti da uvećaju broj izvršenja svake funkcije kako bi se simulirao pristup većeg broja različitih korisnika u odgovarajućem periodu vremena. 3. Sve funkcije koje su vremenski zavisne treba proveriti korišćenjem tačnih i netačnih podataka. 4. Proveriti da li se sve funkcije koje se pojavljuju periodično, izvršavaju/pokreću u odgovarajućem vremenskim trenucima. 5. Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima. 6. Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. 7. Potvrditi da su poslovna pravila pravilno primenjena. |
| Kriterijum završetka: | 1. Svi planirani testovi su izvršeni. 2. Sve identifikovane greške su adekvatno obrađene. |
| Napomene: | 1. Određivanje datuma i perioda može zahtevati dodatne aktivnosti. 2. Poslovni model bi trebao da identifikuje odgovarajuće zahteve i procedure koje treba testirati. |

### Testiranje korisničkog interfejsa (User Interface Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi sledeće:   1. Navigacija kroz aplikaciju korektno odslikava poslovne funkcije i zahteve, uključujući kretanje između prozora, polja i metoda pristupa (prelazak tabom, kretanje miša) 2. Objekti prozora i njihove karakteristike, kao što su meniji, veličina, pozicije, stanja i dobijanje fokusa u skladu sa standardima. |
| Tehnike: | 1. Kreirati/modifikovati testove za svaki prozor da bi se potvrdila odgovarajuća navigacija i stanja objekata za svaki prozor i objekat aplikacije. |
| Kriterijum završetka: | Svaki prozor treba da se potvrdi da je konzistentan sa referentnom verzijom ili prihvaćenim standardima. |
| Napomene: |  |

### Testiranje performansi (Performance Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi vreme odziva aplikacje za napravljene transakcije ili poslovne funkcije za sledeća dva slučaja:  - procenjeno prosečno opterećenje  - procenjeno maksimalno opterećenje |
| Tehnike: | 1. Koristiti test procedura napravljene za testiranje poslovnog modela (testiranje sistema) 2. Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte, da se poveća broj izvršenja transakcija. 3. Skripte treba da se izvršavaju na jednom računaru (jedan korisnik, jedna transakcija) ili ponoviti sa više klijenata (viruelinih ili stvarnih). |
| Kriterijum završetka: | 1. Jedna transakcija / jedan korisnik: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. 2. Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. |
| Napomena: | 1. Testiranje performansi sistema uključuje opterećenje aplikacije u pozadini. Postoji više metoda koji se mogu koristiti da bi se dobili zahtevani uslovi: 2. Pozivati transakcije direktno na serveru, obično u formi Cypher poziva. 3. Kreirati virtuelnog korisnika koji će simulirati više klijenata. Može se iskoristiti softver koji će sa udaljenog računara biti iskorišćen da poveća opterećenje. Time se, takođe, može postići povećanje opterećenja na mreži. 4. Koristiti više ljudi koji će sa različitih računara pokretati test skripte da bi se dostiglo odgovarajuće opterećenje. 5. Testiranje performansi treba da se obavi na za to spremljenom računaru u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje. 6. Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti. |

### Testiranje opterećenja (Load Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi vreme odziva aplikacije za napravljene transakcije i poslovne slučajeve pod uslovima različitog opterećenja. |
| Tehnike: | 1. Koristiti testove napravljene za testiranje poslovnog ciklusa. 2. Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte da se poveća broj izvršenja transakcija. |
| Kriterijum završetka: | 1. Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. |
| Napomene: | 1. Testiranje performansi treba da se obavi na za to spremljenom računaru u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje. 2. Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti. |

### Testiranje kritičnih slučajeva(Stress Testing)

|  |  |
| --- | --- |
| Cilj testa: | Proveriti da li sistem funkcioniše korektno i bez grešaka u sledećim kritičnim slučajevima:   1. malo ili nimalo memorije na serveru 2. maksimalan broj klijenata (stvarnih ili simuliranih) koji rade na sistemu 3. više korisnika koji zahtevaju istu transakciju nad istim podacima. 4. najgori slučaj broja transakcija koje sistem može da izdrži.   Napmena: Cilj testiranja kritičnih slučajeva može biti i određivanje uslova pod kojima sistem nije u mogućnosti da nastavi da radi korektno. |
| Tehnike: | 1. Koristiti testove koji su napravljeni za testiranje performansi. 2. Da bi se obavilo testiranje pod ograničenjima, testovi treba da se obavljaju sa jednog računara, a veličinu RAM-a i spoljne memorije treba smanjiti. 3. Za ostale testove, treba koristiti više klijenata, koji će raditi na istom testu ili na komplementarnim testovima da bi se proizveo slučaj sa najvećim količinom podataka u transakciji. |
| Kriterijum završetka: | Svi planirani testovi se izvršavaju i predviđene granice sistema su dostignute bez pada sistema ili su uslovi pod kojima se pojavljuje otkaz sistema su izvan predviđenih granica. |
| Napomene: | 1. Testiranje mreže može zahtevati korišćenje alata koji omogućavaju povećanje protoka podataka na mreži. 2. Veličina spoljne memorije treba da bude privremeno smanjena kako bi smanjila prostor koji može da koristi baza podataka. 3. Testiranje zahteva sinhronizacije istovremenog pristupa klijenata istim slogovima. |

### Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Sigurnost funkcija/podataka: Potvrditi da korisnik može da koristi samo one funkcije/ podatke koje su predviđene za tip korisnika kome pripada.  Sigurnost sistema: Potvrditi da samo oni korisnici kojima je dozvoljeno da pristupe sistemu mogu da pristupe sistemu. |
| Tehnike: | 1. Sigurnost funkcija/podataka: Identifikovati funkcije/podatke kojima ima pravo pristupa odgovarajući tip korisnika. 2. Napraviti testove za svaki tip korisnika i potvrditi dozvolu da izvršava transakcije koje su specifične za taj tip korisnika. 3. Izmeniti tip korisnika i proveriti ga na iste testove. Proveriti da li su funkcije /podaci korektno dozvoljeni ili zabranjeni za korišćenje. |
| Kriterijum završetka: | Za svaki tip korisnika i odgovarajuće funkcije/podatke dozvoljene za korišćenje, sve transakcije funkcionišu na očekivani način, definisane u testovima sistema. |
| Napomene: |  |

### Testiranje konfiguracije (Configuration Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Proveriti i potvrditi da klijentska aplikacija funkcioniše na konfiguracijama na kojima je to predviđeno. |
| Tehnike: | 1. Koristi testove integracije i funkcionalne testove. 2. Pokrenuti/zatvoriti različite aplikacije, bilo pre pokretanja testa, bilo kao deo testiranja. 3. Izvršiti izabrane transakcije da bi se simulirale korisničke aktivnosti u različitim aplikacijama. 4. Ponoviti navedene procese, smanjujući RAM memoriju koju koristi klijent. |
| Kriterijum završetka: | Za svaku kombinaciju, transakcije iz testa treba da se završe uspešno, bez grešaka. |
| Napomene: | 1. Koje su aplikacije dostupne i mogu se koristiti na klijentu? 2. Koje se aplikacije tipično koriste? 3. Koje podatke aplikacija koristi (koji je obim tih podataka) 4. Ceo sistem, mrežni serveri, baze podataka i slično treba biti opisano kao deo ovog testa. |

## Alati

U sledećoj tabeli su pobrojani alati koji će biti korišćeni za testiranje sistema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Faza** | **Alat** | **Proizvođač/ matični proiz.** | **Verzija** |
| Test Management | Rational Unified Process | Rational |  |
| Test Design | Rational Rose | Rational |  |
| Defect Tracking |  |  |  |
| Functional Testing |  |  |  |
| Performance Testing |  |  |  |
| Test Coverage Monitor or Profiler |  |  |  |
| Other Test Tools |  |  |  |
| Project Management | Project  Word  Excel | Microsoft  Microsoft  Microsoft | Office 365 |
| DBMS tools | Neo4j | Neo4j | 4.4.2 |

# 

# Resursi

U ovom odeljku su opisani resursi koji se mogu koristiti za testiranje sistema, glavne odgovornosti i njihovo znanje ili veštine.

## Radnici

Ova tabela pokazuje predlog rasporeda radnika na aktivnostima testiranja.

|  |  |  |
| --- | --- | --- |
| **Ljudski resursi** | | |
| **Radnik** | **Preporučeni minimalni broj radnika**  (broj radnika koji će raditi puno radno vreme) | **Posebne preporuke / komentari** |
| Rukovodilac testiranja | Dušan | Obezbeđuje nadgledanje upravljanjem procesa  Odgovornosti:   1. Obezbediti tehničku direkciju 2. Prikupljanje odgovarajućih resursa 3. Upravljanje izveštavanjem |
| Projektant testova | Dušan | Identifikuje, određuje prioritete i implementira slučajeve testiranja  Odgovornosti:   1. Napraviti plan testiranja 2. Napraviti model testiranja 3. Proceniti efektivnost napora uloženog u testiranje. |
| Tester sistema | Dušan | Izvršiti testiranje  Odgovornosti:   1. Izvršiti testiranje 2. Zabeležiti rezultate 3. Popraviti greške 4. Opisati greške |
| Administrator testiranja sistema | Dušan | Pripremiti okruženje za testiranje i utvrditi da ono odgovara pretpostavkama.  Odgovornosti:   1. Administrirati upravljanje testiranjem 2. Upravljati pristupom radnika resursima na kojima se obavlja testiranje. |
| Administrator baze podataka | Dušan | Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.  Odgovornosti:   1. Administrirati podatke (bazu podataka) za testiranje |
| Projektant | Dušan | Identifikovati i definisati operacije, atribute i veze između različitih klasa testova  Odgovornosti:   1. Identifikovati i definisati klase testova 2. Identifikovati i definisati pakete testova |
| Implementacija | Dušan | Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.  Odgovornosti:   1. Kreirati klase i pakete testova implementirane u test modelu. |

## Sistem

U sledećoj tabeli su navedeni sistemski resursi koji će se koristiti za testiranje TaF sistema.

|  |  |
| --- | --- |
| **Sistemski resursi** | |
| **Resurs** | **Ime / Tip / Serijski broj** |
| Server | Ime: D5011D01-B0A3-4510-A5DB-A3592DA55987 |
| Klijentski Test Računari |  |
| —2 udaljena računara  (sa internet pristupom) | Ime: D5011D01-B0A3-4510-A5DB-A3592DA55987  Ime: DESKTOP- B0C365A5 |
| 1 lokalni računar (povezanih u lokanu mrežu) | Ime: DESKTOP- D5011D01-B0A3-4510-A5DB-A3592DA55987 |

# Kontrolne tačke testiranja

Aktivnosti testiranja zavise od iteracija u razvoju softvera. Ciklus razvoja softvera TaF biće odrađen u jednoj iteraciji koja sadrži kompletan ciklus testiranja, planiranje, projektovanje, razvoj, izvršenje i evaluaciju.

Sledeća tabela prikazuje kontrolne tačke testiranja.

|  |  |  |  |
| --- | --- | --- | --- |
| **Zadatak** | **Trud (pd)** | **Početak** | **Kraj** |
| Iteracija C1: R1.0 Release  Planiranje testova  Projektovanje testova  Razvoj testova  Izvršenje testova  Evaluacija testova | 5 | Maj 18 | Maj 25 |

# Izlazni produkti testiranja

Rezultati testiranja koji su definisani ovim planom testiranja navedeni su u sledećoj tabeli.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument** | **Vlasnik** | **Ocena / Raspodela** | **Due Date** |
| Plan Testiranja | Dušan | Rukovodioci testiranja | Maj 18 |
| Test okruženje | Dušan | - | Maj 19 |
| Model testiranja | Dušan | Interna ocena | Maj 20 |
| Skup test podataka | Dušan | Interna ocena | Maj 21 |
| Test procedure | Dušan | Interna ocena | Maj 22 |
| Test skripte | Dušan | - | Maj 23 |
| Test podrška | Dušan | - | Maj 23 |
| Izveštaj o uočenim greškama | Dušan | Rukovodioci testiranja | Maj 24 |
| Rezultati testa | Dušan | Rukovodilac testiranja | Maj 24 |
| Izveštaj o evaluaciji testa | Dušan | Rukovodilac testiranja | Maj 25 |

## Model testiranja

Model testiranja definiše sve test slučajeve zajedno sa referencama na test procedure i test skripte koje su vezane za svaki test slučaj.

# Projektni zadaci

U sledećoj tabeli su navedeni projektni zadaci testiranja sistema:

|  |
| --- |
| **Plan testiranja** |
| Identifikovati zahteve za testiranjem |
| Proceniti rizike |
| Razviti strategije testiranja |
| Identifikovati resurse testiranja |
| Napraviti raspored |
| Napraviti Plan testiranja |
| **Projektovanje testova** |
| Analizirati potrebni rad |
| Razviti model testiranja |
| Identifikovati i opisati test slučajeve |
| Identifikovati i grupisati procedure testiranja |
| Oceniti napraviti uvid u pokrivenost testiranja |
| **Implementacija testova** |
| Postaviti okruženje za testiranje |
| Napisati skripte za testiranje |
| Razviti podršku za testiranje |
| Identifikovati sprecifične funkcionalnosti za projektovanje i implementaciju modela testiranja |
| Utvrditi eksterne skupove podataka |
| **Testiranje** |
| Izvršiti test procedure |
| Oceniti izvršenje testova |
| Popraviti testove koji nisu prošli |
| Proveriti rezultate |
| Istražiti neočekivane rezultate |
| Zabeležiti greške |
| **Ocena testiranja** |
| Oceniti pokrivenost funkcionalnosti test slučajevima |
| Proceniti pokrivenost koda test slučajevima |
| Analizirati greške |
| Utvrditi da li su kriterijumi završetka i kriterijumi uspešnosti postignuti |
| Napraviti Izveštaj o oceni testiranja |