Cogit

Plan testiranja

Verzija 1.0

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 11.05.2021. | 1.0 | Inicijalna verzija | AmanSoft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Ciljevi 4

2. Opseg 4

3. Reference 4

4. Zahtevi za testiranje 4

5. Strategije testiranja (Test Strategy) 6

5.1 Tipovi testiranja (Testing Types) 6

5.1.1 Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing) 6

5.1.2 Testiranje sistema (System Testing) 7

5.1.3 Testiranje poslovnog ciklusa (Business Cycle Testing) 7

5.1.4 Testiranje korisničkog interfejsa (User Interface Testing) 8

5.1.5 Testiranje performansi (Performance Testing) 8

5.1.6 Testiranje opterećenja (Load Testing) 9

5.1.7 Testiranje kritičnih slučajeva(Stress Testing) 9

5.1.8 Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing) 10

5.1.9 Testiranje konfiguracije (Configuration Testing) 10

5.2 Alati 11

6. Resursi 11

6.1 Radnici 11

6.2 Sistem 13

7. Kontrolne tačke testiranja 13

8. Izlazni produkti testiranja 13

8.1 Model testiranja 14

9. Projektni zadaci 14

Plan Testiranja

# Ciljevi

Ovaj dokument sadrži opis plana testiranja Cogit sistema. Namena ovog dokumenta je da opiše način provere funkcionalni i nefunkcionalne zahteve postavljene 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 testiranje.
* Navesti dokumente koji će opisati rezultate testiranja.

# Opseg

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

Plan testiranja ne opisuje testiranje svake pojedinačne komponenti aplikacije, jer se pretpostavlja se 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 Cogit aplikacije,
2. Specifikacija zahteva Cogit aplikacije i
3. Detaljno – arhitekturni projekat Cogit.

# Zahtevi za testiranje

U ovom odeljku biće navedeni oni elementi sistema 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 Cogit aplikacije.

Proveriti mogućnost istovremenog pristupa istim podacima radi čitanja.

Proveriti tačnost podataka koji se pribavljaju iz baze.

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

Proveriti slučaj korišćenja Kreiranje naloga.

Proveriti slučaj korišćenja Prijavljivanje na sistem.

Proveriti slučaj korišćenja Podešavanje naloga.

Proveriti slučaj korišćenja Kreiranje tima.

Proveriti slučaj korišćenja Komunikacija sa ostalim korisnicima.

Proveriti slučaj korišćenja Korišćenje kalendara.

Proveriti slučaj korišćenja Napuštanje tima.

Proveriti slučaj korišćenja Prikaz projekata.

Proveriti slučaj korišćenja Upravljanje zadacima.

Proveriti slučaj korišćenja Arhiviranje i brisanje zadataka.

Proveriti slučaj korišćenja Kreiranje i dodeljivanje zadataka članovima.

Proveriti slučaj korišćenja Dodavanje prioriteta zadatku.

Proveriti slučaj korišćenja Upravljanje projektima.

Proveriti slučaj korišćenja Upravljanje timom.

Proveriti slučaj korišćenja Suspendovanje korisnika.

Proveriti slučaj korišćenja Brisanje korisnika.

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

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

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

Proveriti posledice operacije dodavanja novog člana tima.

Proveriti posledice operacije arhiviranja postojećeg projekta.

Proveriti posledice operacije definisanja novog projekta.

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

Proveriti jednostavnost navigacije kroz skup prozora aplikacije.

Proveriti da li korisnički interfejs omogućava jednostavno i intuitivno korišćenje bez potrebe za organizovanjem dodatne obuke.

Sistem treba da bude lak za korišćenje i prilagođen korisnicima koji poseduju dobar nvo znanja rada na računaru.

Proveriti da li *online* uputstvo obezbeđuje adekvatnu podršku pri korišćenju funkcionalnosti sistema .

**Testiranje performansi (Performance Testing)**

Proveriti vreme potrebno za prijavljivanje na sistem.

Proveriti vreme potrebno za dodavanje novog projekta.

Proveriti da li vreme potrebno za pristupanje bazi podataka u cilju izvršenje upita ne prelazi 5 sekundi.

**Testiranje opterećenja (Load Testing)**

Proveriti odziv sistema kada mu istovremeno pristupa 50 posetilaca.

**Testiranje kritičnih slučajeva (Stress Testing)**

Proveriti vreme odziv sistema prilikom prvog korišćenja.

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

Proveriti prava pristupa korisnika koji pripadaju različitim grupama (Posetilac aplikacije, Član Tima, Vođa Tima, Administrator).

**Testiranje konfiguracije (Configuration Testing)**

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

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

Proveriti da li klijentski deo Cogit aplikacije može da se koristi pomoću Web čitača: Opera 8.0 i noviji, Firefox (Mozilla), kao i Google Chrome.

# 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 na 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. Svi 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 aplikacija 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 da se 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. Svi identifikovane greške su adekvatno obrađene. |
| Napomene: | 1. Određivanje datuma i perioda može zahtevati dodatne aktivnosti. 2. Poslovni model bi trebao ad 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 od kretanje između prozora, polja i metode 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 sistema za napravljene transakcije ili poslovne funkcije za sledeća dva slučaja:  - procenjeno prosečno opterećenje  - procenjeno maksimalno opterećenje |
| Tehnike: | 1. Koristiti test procedure 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 sistema u pozadini. Postoji više metoda koji se mogu koristiti da bi se dobili zahtevani uslovi: 2. Pozivati transakcije direktno na serveru. 3. Koristiti više ljudi koji će sa različitih računara pokretati test skripte da bi se dostiglo odgovarajuće opterećenje. 4. 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 sistema 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. 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. maksimalan broj klijenata (stvarnih ili simuliranih) koji rade na sistemu   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. |
| 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. |

### 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. Izvršiti izabrane transakcije da bi se simulirale korisničke aktivnosti u različitim aplikacijama. |
| Kriterijum završetka: | Za svaku kombinaciju, transakcije iz testa treba da se završe uspešno, bez grešaka. |
| Napomene: | 1. Koje se funkcije tipično koriste? 2. Koje podatke aplikacija koristi (koji je obim tih podataka) |

## 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 | TBD |
| Test Design | Rational Rose | Rational | TBD |
| Defect Tracking |  |  | TBD |
| Functional Testing |  |  | TBD |
| Performance Testing |  |  | TBD |
| Test Coverage Monitor or Profiler |  |  | TBD |
| Other Test Tools |  |  | TBD |
| Project Management | Word | Microsoft | TBD |
| DBMS tools | MongoDB | MongoDB Inc. | 4.4.4 |

# 

# Resursi

U ovom odelju 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 | Mateja, Aleksandar | Obezbeđuje nadgledanje upravljanjem procesa  Odgovornosti:   1. Obezbediti tehničku direkciju 2. Prikupljanje odgovarajućih resursa 3. Upravljanje izveštavanjem |
| Projektant testova | Mateja, Aleksandar | 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 | Nikola,  Ana | Izvršiti testiranje  Odgovornosti:   1. Izvršiti testiranje 2. Zabeležiti rezultate 3. Popraviti greške 4. Opisati greške |
| Administrator testiranja sistema | Aleksandar | 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 | MatejaC | Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.  Odgovornosti:   1. Administrirati podatke (bazu podataka) za testiranje |
| Projektant | Mateja | 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 | Ana  Nikola | Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.  Odgovornosti:   1. Kreirati klase i pakete testova implementirane u test modelu. |

## Sistem

Sistemski resursi su računari članova tima.

# Kontrolne tačke testiranja

Aktivnosti testiranja zavise od iteracija u razvoju softvera. Ciklus razvoja Cogit softvera 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 11 | Maj 23 |

# Izlazni produkti testiranja

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument** | **Vlasnik** | **Ocena / Raspodela** | **Due Date** |
| Plan Testiranja | AmanSoft | Rukovodioci testiranja | Maj 14 |
| Test okruženje | AmanSoft | - | Maj 14 |
| Model testiranja | AmanSoft | Interna ocena | Maj 14 |
| Skup test podataka | AmanSoft | Interna ocena | Maj 16 |
| Test procedure | AmanSoft | Interna ocena | Maj 17 |
| Test skripte | AmanSoft | - | Maj 17 |
| Test podrška | AmanSoft | - | Maj 17 |
| Izveštaj o uočenim greškama | AmanSoft | Rukovodioci testiranja | Maj 23 |
| Rezultati testa | AmanSoft | Rukovodilac testiranja | Maj 23 |
| Izveštaj o evaluaciji testa | AmanSoft | Rukovodilac testiranja | Maj 23 |

## 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 |