

*Prolećni semestar, 2021/22*

*SE325 - Upravljanje projektima razvoja softvera*

**Razvoj inovativnog rasporeda časova**

Projektni zadatak 2

Profesor: **Nemanja Zdravković**

Asistent: **Tamara Vukadinović**

Student: **Aleksa Cekić 4173**

Niš, 2022.

**Sadržaj**

[Uvod 3](#_Toc105511904)

[Apstrakt sistema 3](#_Toc105511905)

[Postupak razvoja softvera – teorijska osnova 4](#_Toc105511906)

[SDLC na konkretnom primeru 4](#_Toc105511907)

[Zaključak 5](#_Toc105511908)

[Literatura 5](#_Toc105511909)

# Uvod

Životni ciklus razvoja softvera ili SDLC je postupak koji se koristi za razvoj softvera. Postoje različite faze ili faze u životnom ciklusu razvoja softvera i u svakoj fazi se odvijaju različite aktivnosti.

SDLC stvara strukturu za razvojne timove kako bi mogli dizajnirati, kreirati i isporučivati ​​visokokvalitetni softver definiranjem različitih zadataka koji se trebaju dogoditi Životni ciklus defineše metodologiju za poboljšanje kvaliteta softvera i celokupnog procesa razvoja.

Namera SDLC-a je da pomogne u proizvodnji proizvoda koji je isplativ, delotvoran i visokokvalitetan.

# Apstrakt sistema

Krajnji cilj aplikacije jeste sistem koji omogućava pravljenje rasporeda časova na lep, organizovan način, sa raznim alatima koji će pomoći korisniku da vodi računa o vremenu. Jedan od problema koji je inače zastupljen u društvu je taj da su ljudi, pogotovo studenti, uglavnom neorganizovani što im otežava prilikom studiranja i spremanju ispita. Aplikacija je namenjena da korisniku omogući pregled svih potrebnih informacija o predmetu, kao i vremenu početka i završetka predavanja i vežbi tih predmeta koje korisnik unosi u sistem.

Postojećih rešenja već ima u vidu informacionih web sajtova gde korisnici pored ostalog imaju i pristup rasporedu časova, ali sa manjim fokusom na raspored. Sistem koji se razvija bi pružio set interaktivnih alata kao što su alarmi za početak nastava, pristup online predavanjima preko video strima (stream) sa četom (chat) na nivou unetih predmeta. Ovakav vid aplikacije, obzirom na to da se radi o multi-funkcionalnoj aplikaciji, pruža to da se korisnici na organizuju na bolji i efektivniji način.

# Postupak razvoja softvera – teorijska osnova

SDLC se se stoji od sledećih aktivnosti:

1. **Planiranje:** Najvažnije dijelove razvoja softvera, prikupljanje zahtjeva ili analizu zahtjeva obično izvode najvještiji i iskusniji softverski inženjeri u organizaciji. Nakon što se od klijenta prikupe zahtjevi, izrađuje se dokument s opsegom u kojem se utvrđuje i dokumentuje opseg projekta.
2. **Implementacija:** Softverski inženjeri počinju sa pisanjem koda prema zahtevima klijenta.
3. **Ispitivanje:** ovo je postupak pronalaženja grešaka ili grešaka u kreiranom softveru.
4. **Dokumentacija:** Svaki korak u projektu dokumentuje se za buduću referencu i za poboljšanje softvera u procesu razvoja.
5. **Uvođenje i održavanje:** softver se primenjuje nakon odobrenja za puštanje u promet.
6. **Održavanje:** Održavanje softvera izvodi se za buduću referencu. Poboljšanje softvera i novi zahtevi (zahtevi za promjenom) mogu potrajati duže od vremena potrebnog za stvaranje početnog softvera.

# SDLC na konkretnom primeru

**1. Analiza i definisanje zahteva:** Zahtevan je razvoj aplikacije koja je namenjena da korisniku omogući pregled svih potrebnih informacija o predmetu, kao i vremenu početka i završetka predavanja i vežbi tih predmeta koje korisnik unosi u sistem. Sistem koji se razvija bi pružio set interaktivnih alata kao što su alarmi za početak nastava, pristup online predavanjima preko video strima (stream) sa četom (chat) na nivou unetih predmeta.

,

**2.Projektovanje sistema i softvera :** Projektovanje sistema i softvera u kome se zahtevi raspoređuju svim komponentama sistema i uspostavlja se celokupna arhitektura sistema i to kroz gore navedene dijagrame strukture i dijagrame ponašanja, stanja, aktivnosti...

**3. Implementacija i testiranje jedinica:**U ovoj fazi je projektno rešenje softvera realizovano skupom programa ili programskih jedinica i to na različitim platformama odnosno web-u i aplikaciji. Testiranje jedinica utvrđuje da li svaka jedinica posebno ostvaruje svoju planiranu funkciju.

**4. Integracija i testiranje sistema:** Sve programske jedinice u ovoj fazi se integrišu u sistem i testiraju se kao kompletan sistem radi provere da li softver zadovoljava postavljene zahteve,tj. ostvaruje svoje funkcije i performanse. Posle ovog testiranja, softverski sistem se pušta u komercijalu upotrebu.

**5. Operativni rad i održavanje:** Sistem je instalisan i pušten u operativni rad, tj. u upotrebu. Održavanje obuhvata ispravljanje grešaka koje nisu otkrivene u ranijim fazama životnog ciklusa, ali ostavlja i mogućnost unapređenja sistema kao npr. razvoj funkcionalnosti za specifične naručioce.

# Zaključak

Sve gore definirane faze u životnom ciklusu razvoja softvera primenjive su na bilo koju metodologiju razvoja softvera, ali trajanje i aktivnosti u svakoj fazi zavise od toga da li koristimo V model metodologije razvoja ili Agilni pristup.

U Agilnom pristupu je trajanje isporuke radnog softvera obično između 2 i 4 sedmice, pa se svaka od gore navedenih faza skraćuje. Takođe agilno, to je celovit timski pristup u koji su uključeni programeri i testeri koji zajedno rade na stvaranju visokokvalitetnog proizvoda.

# Literatura

* <https://bs.invest-map-nnov.com/software-development-life-cycle-sdlc>