



# Principi softverskog inženjerstva

## Vežbe - I nedelja - Uvod

Dražen Drašković, asistent  
Elektrotehnički fakultet  
Univerziteta u Beogradu



- Naziv predmeta:  
Principi softverskog inženjerstva
- obavezan predmet
- Godina: III, Semestar: 6
- Fond časova: 2 + 2 + 1
- Broj ESPB bodova: 6
- Preduslov: nema

# Nastavnici

- Predavanja: ponedjeljak, 10<sup>00</sup>-12<sup>00</sup>  
Doc. dr Dragan Bojić, kancelarija P23  
bojic@etf.rs
  - Vežbe: četvrtak, 10<sup>00</sup>-12<sup>00</sup>  
dipl.ing. Dražen Drašković, kancelarija 37  
drazen.draskovic@etf.rs
- Konsultacije posle nastave i po dogovoru.

# Informisanje studenata



- Predmetni sajt:  
<http://si3psi.etf.rs>
- Mejling lista:  
[si3psi@lists.etf.rs](mailto:si3psi@lists.etf.rs)
  - Nakon što prijavite predmet, bićete automatski dodati sa studentskom e-mail adresom!
  - Sa drugim, privatnim e-mail adresama nije moguć pristup listi!

# Pravila polaganja ispita



- 40% poena kroz projekat koji se radi u toku semestra
- 60% poena pismeni ispit (K1+K2+I)
- Na pismenom ispitu (i kolokvijumima) u obzir dolazi gradivo sa predavanja, vežbi i pitanja u vezi sa projektom

# Pismeni ispit

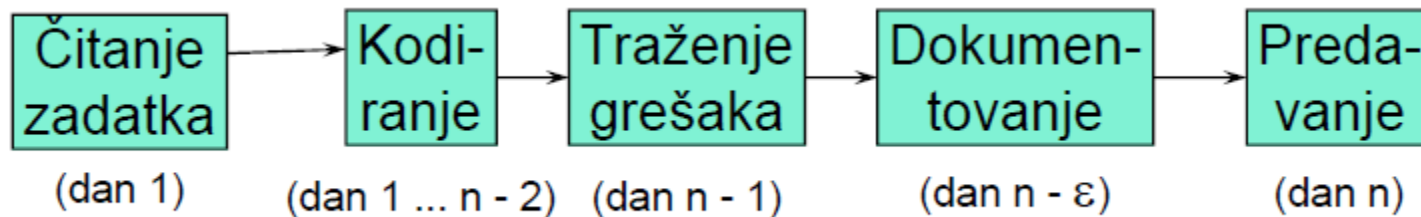
- U svakom ispitnom roku student može polagati ispit koji pokriva celo gradivo i boduje se sa 60 poena. Ukoliko je student polagao kolokvijume uzima se najbolja varijanta zaključivanja ocena:
  - ako se računaju oba kolokvijuma:  
**Total = PRO + K1 + K2 + ispit \* 1/3**
  - ako se računa jedan od kolokvijuma:  
**Total = PRO + Kx + ispit \* 2/3**
  - ako se računa samo ispit:  
**Total = PRO + ispit \* 1**
- Uslov za pozitivnu ocenu je da ukupan broj poena koji se osvoji kroz kolokvijume i završni ispit  $\geq 30$

# Projekat



- Projekat je obavezan i nosi 40 poena.
- Projekat može da se radi samo za junijski ispitni rok, u grupama od najviše 4 člana i usmeno se brani u unapred određenim terminima pre početka ispitnog roka.
- Jednom odbranjeni projekat važi u svim ispitnim rokovima tekuće školske godine.
- Napomena: samo u JUNU!!!

# Studentski razvoj softvera ☺



Studentski pogled na životni vek softvera!



# Terminologija u softverskom inž.

- Softver
- Softversko inženjerstvo
- Software Engineering vs Computer Science
- Software Engineering vs System Engineering
- Proceniti softver

# Ovo su rekle vaše kolege o projektu...

- “Neki delovi su bili monotoni, a većina krajnje interesantni. Na kraju kad prođeš sve faze, vidiš da i nije loše to što uradiš.”
- “Projekat je obiman, ali zanimljiv, pa mu sa pravom treba posvetiti toliku pažnju.”
- “Mislím da je najvažnija stvar na celom predmetu projekat. Najbolji od svih projekata.”
- “Na vežbama se posvećivalo dosta vremena projektu i opisivanju faza, možda čak i svo vreme, ali i pored toga ja tu ništa ne bih menjao, što kaže Zdravko Čolić.”

# Projekat

## Faze

- **Izrada projektnog zadatka**
- Faza definisanja zahteva (specifikacija+ prototip)
- Faza formalne inspekcije
- Faza modelovanja podataka
- Faza implementacije (veb aplikacija + cloud platforme)
- Faza razvoja testova
- Dokumentovanje i (re)dokumentovanje
- ...

# Projektni zadatak



- Uvod
  - ukratko šta se radi u projektu
  - namena dokumenta
  - ciljevi projekta
- Opis sistema koji se realizuje
- Kategorije korisnika sistema
- Tehnički koncept rešenja
- Plan razvoja sistema i prioriteti
- Vizija (pretpostavke, kontraindikacije, alternative,...)

# Projektni zadatak

## Tehnički koncept rešenja

- Pregled arhitekture sistema
- Pregled karakteristika
- Dodatni zahtevi

# Projektni zadatak

## Specifikacija korisničkih zahteva

- Specifikacija korisničkih zahteva je zvaničan dokument šta treba implementirati. Ona treba da obuhvati i zahteve korisnika, ali i sistemske zahteve.
- Ako ima veliki broj zahteva, oni mogu da čine i poseban dokument.
- Raznolikost mogućih korisnika

# Projektni zadatak

## Specifikacija korisničkih zahteva

- Funkcionalni zahtevi - osnovne funkcije koje sistem treba da obezbedi različitim kategorijama korisnika
- Nefunkcionalni zahtevi
  - Sistemski zahtevi
  - Ostali zahtevi
- Zahtevi za korisničkom dokumentacijom

# IEEE standard (1998)

## Specifikacija korisničkih zahteva

- Uvod
- Opšti opis
  - Korisnici
  - Funkcije sistema
  - Opšta ograničenja
  - Pretpostavke i zavisnosti
- Specifični zahtevi
- Dodaci
- Index



# Softverski timovi (1)



- Nekoliko pojedinaca dobilo je 2 različita funkcionalna zadatka, zajednički rad relativno retko;
- Koordinacija je odgovornost tim lidera koji može da vodi još 5-6 drugih projekata;

# Softverski timovi (2)



- Nekoliko pojedinaca dobilo je 2 različita funkcionalna zadatka, tako da su uspostavljeni neformalni timovi;
- Ad hoc vođa tima može biti imenovan, ali i ne mora;

# Softverski timovi (3)



- Nekoliko pojedinaca organizovano je u 2 tima;
- Svaki tim je dobio jedan ili više funkcionalnih zahteva; svaki tim ima određenu strukturu koja je definisana za svaki tim koji radi na projektu;
- Koordinacija je odgovornost oba tima i glavnog menadžera projekta;
- Primer Računskog centra ETF-a:
  - tim programera koji radi EVIDES (eStudent, eZaposleni)
  - tim programera koji radi FIMES
  - zajednički projekat FIS = Fakultetski IS



- MOI model rukovođenja:
  - motivacija - sposobnost lidera da podstakne članove tima da pruže svoj maksimum
  - organizacija - sposobnost lidera da od početnog koncepta realizuje finalni produkt
  - ideje ili inovacije - sposobnost lidera da podstakne članove tima da budu kreativni, čak i kada moraju da rade u okviru granica uspostavljenih za određeni softverski produkt

# Još neke osobine tim lidera

- rešavanje problema
- preuzimanje kontrole kada je potrebno
- sposobnost da kod člana tima “pročita” signale (verbalna i neverbalna komunikacija)
- kontrola situacije

# Projekat - I faza



Pa da započnemo...

*projekat*

# Projekat - I faza



- Formiranje timova

Studenti daju ime svom timu i biraju jednog tim lidera. Vođa tima je zadužen za komunikaciju sa predmetnim asistentom oko svih pojedinosti projekta.

Individualni rad nije zabranjen, ali se ne preporučuje.

# Projekat - Formiranje timova

- Krajnji rok za podelu u timove je (ponedeljak) **02.03.2015. u 23:59**
- Do tada je potrebno na adresu *drazen.draskovic@etf.rs* poslati mejl sa subject-om: TIM <odabrano\_ime\_tima>
- U telu mejla navesti:
  - ime i prezime, broj indeksa i e-mail svakog člana tima
  - navesti ko je vođa tima
  - voditi računa da se pošalju ispravne e-mail adrese!



# Projekat - Definisiranje projektnog zadatka

- Potrebno je da svaki tim definiše svoj projektni zadatak na osnovu priloženog primera (može se slati ili .doc ili .pdf fajl)
- Kao subject potrebno je navesti TIM <ime\_tima> PZ verzija X.Y
- Projektni zadatak piše vođa tima i eventualno još jedan član tima (uz konsultacije sa drugim članovima tima).

Krajnji rok za slanje projektnog zadatka je **09.03.2015. (ponedeljak) u 23:59.**

# Projekat - I faza



- Dobra praksa kroz projekat
- Aplikaciju uvek treba uskladiti sa željama kupca (funkcionalnost, cena, rokovi,...)
- Na ovom primeru: asistent je kupac!

# Agilne metodologije razvoja (1)

- Nastale početkom 21. veka
- „Manifest agilnog softverskog razvoja“, objavljen od strane 17 softverskih inženjera (februar 2001.)
- 12 principa agilnog softverskog razvoja:
  - **Pojedinci i interakcije** ispred procesa i alata
  - **Softver koji radi** nasuprot opsežnoj dokumentaciji
  - **Saradnja sa klijentima** nasuprot ugovornim aranžmanima
  - **Prilagođavanje promenama** nasuprot pravljenju planova

# Agilne metodologije razvoja (2)

- Projekti - treba da budu razvijani od strane visoko motivisanih članova tima; treba im omogućiti okruženje i podršku koja im je neophodna.
- Timovi treba da u redovnim intervalima razmatraju kako da postanu efikasniji i da na osnovu razmene znanja i iskustva usklađuju svoje postupke.

# Agilne metodologije razvoja (3)

- Preuzete ideje iz XP (ekstremnog programiranja):
  - „programiranje u paru“ - za najsloženije delove sistema
  - „neprekidna integracija“ - praviti mala unapređenja i poboljšanja sistema i posle svake takve promene omogućiti da sistem funkcioniše kao potpuna celina
- Ideje iz metodologije „Rapidnog razvoja aplikacija“: izrada prototipa
- Dinamičko prilagođavanje promenama
- I sama metodologija može doživeti određene promene, ako se u praksi pokažu kao dobre

# Osnovne ideje i saveti (1)

- Promovisanje direktnog i timskog rada
- Akcenat na visokoj motivisanosti svih članova tima, njihovom dobrom međusobnom poznavanju poštovanju i uvažavanju
- Članovi tima treba da inspirišu jedni druge - da guraju jedni druge napred
- Iz jedne realizovane ideje treba da se rodi 10 novih ideja, i da na taj način broj ideja raste
- Zapisati svaku ideju koliko god ona loše zvučala, preispitati svaku ideju koliko god ona dobro zvučala

# Osnovne ideje i saveti (2)



- Komunikacija između članova tima treba da teče direktno i neposredno, a ne preko dokumentacije - dokumentovati samo bitne i složene delove sistema
- Paralelno sa razvojem vršiti i najosnovnija testiranja
- Na početku definisati samo osnovne funkcionalnosti, bez kojih sistem ne bi imao smisla, a kasnije dodavati nove napredne funkcionalnosti

# Osnovne ideje i saveti (3)

- Svaki deo sistema treba da bude što je moguće više modulara, kako bi se lako mogao zameniti i dodati novi
- Ako se naiđe na problem koji je moguće rešiti na više načina, u dogovoru sa timom pronaći najbolje rešenje, dokumentovati ga i primeniti ga svaki put kada se naiđe na isti ili sličan problem
- Neke karakteristične stvari dobro bi bilo da se prethodno primene i na stil programiranja
- Kada se razvija neki kritičan ili složen deo sistema, primeniti programiranje u paru ili čak više članova



# Principi koje treba primeniti po završetku svake celine (1)

- Ukratko objasniti ostatku tima šta je i kako urađeno
- Pogledati da li rešenje može biti:
  - brže i efikasnije u pogledu performansi
  - jednostavnije
- Pogledati da li su ispoštovane sve funkcionalnosti vezane za taj deo
- Pogledati koje realne promene bi mogle biti zahtevane u budućnosti, i da li bi ih bilo lako napraviti

# Principi koje treba primeniti po završetku svake celine (2)

- Testirati najosnovnije funkcionalnosti realizovanog dela
- Pogledati da li se neki deo može izdvojiti kako bi bio reupotrebljen
- Najbitnije da se sve ove tačke razmatraju u prisustvu celog tima, gde će svako izneti po neki predlog i komentar

# Principi koji se odnose na tim (1)

- Broj članova tima ne treba biti preveliki, kako bi neposredna komunikacija i organizacija bile moguće i kvalitetne
- Svaki član tima mora biti posvećen projektu i visoko motivisan
- Dobro je ako članovi tima razmišljaju na različite načine, kako bi se razmotrile sve moguće opcije, ali je takođe dobro ako članovi tima rade konzistentno i u „sličnom stilu“
- Loše je ako se neki član tima konstanto bavi istim stvarima (monotonija, manje kvalitetno)

# Principi koji se odnose na tim (2)

- Loše je i ako se neki član tima stalno prebacuje sa jedne na drugu aktivnost
- Od vitalne je važnosti da članovi tima neprekidno razmenjuju znanja i iskustva - kako ona sa kojima ulaze u projekat, tako i ona koja stižu radeći na projektu
- Dobro je u svaki aspekt razvoja ubaciti malo kreativnosti ili neku individualnu crtu, ali naravno ništa od navedenog ne sme biti u suprotnosti sa osnovnom idejom

# Zaključak



- Samo zadovoljan radnik može biti dobar radnik!
- Rad na projektu treba shvatiti kao deo sebe, a ne kao obavezu - sve pod pretpostavkom da je projekat dovoljno inspirativan da tako nešto omogućava.