



# Principi softverskog inženjerstva

## **Vežbe - III nedelja** **Planiranje projekta**

Dražen Drašković, asistent  
Elektrotehnički fakultet  
Univerziteta u Beogradu

# Modeli životnog ciklusa softvera

## Podela

- Model vodopada (eng. *waterfall*)
- Spiralni model (eng. *spiral*)
- Inkrementalni model (eng. *incremental*)
- Iterativni model (eng. *iterative*)
- Agilne metodologije (eng. *agile development*)
  - Ekstremno programiranje (XP)
  - Metod razvoja dinamičkih sistema
  - SCRUM

# Zadatak 1



- Neka kompanija želi da unapredi svoj softverski proizvod. Funkcionalnosti koje treba da budu dodate opisane su u 5 slučajeva korišćenja. Napor koji je potrebno uložiti, izražen u čovek/danima, dat je u zagradi, pored svake faze:
  - Dizajniranje jednog slučaja korišćenja (6 č/d)
  - Implementacija jednog slučaja korišćenja (6 č/d)
  - Pripremanje testova za jedan slučaj korišćenja (1 č/d)  
Napomena: kompletan dizajn slučaja korišćenja mora postojati pre početka pripreme testa
  - Testiranje jednog slučaja korišćenja (1č/d)
  - Integracija jednog slučaja korišćenja u postojeći sistem (1č/d)  
Napomena: ova faza obuhvata i integraciono testiranje
  - Upravljanje projektom (1č/d svake nedelje tokom čitavog trajanja projekta)

# Zadatak 1 (nastavak)



- Tim koji realizuje projekat čini 7 članova:
  - 1 menadžer projekta (zadužen za upravljanje projektom)
  - 1 softverski dizajner
  - 2 programera (mogu da rade dizajn, ukoliko je potrebno)
  - 1 tester (koji priprema i pokreće testove)
  - 1 integrator
- Napraviti projektni plan i predstaviti sve aktivnosti u Gantt dijagramu, za dva različita modela procesa:
  - a) Projekat koristi model vodopada (odnosno svi slučajevi upotrebe se obrađuju zajedno u svakoj fazi)
  - b) Projekat koristi inkrementalni model (odnosno slučajevi korišćenja se dodaju u sistem po jedan u svakom trenutku)

Ukoliko je moguće, cilj je smanjiti ukupno vreme, identifikovanjem aktivnosti koje mogu da se preklapaju. Za svaki model, odrediti ukupno kalendarsko vreme trajanja projekta.



# Zadatak 1 - Rešenje (a)

## Model vodopada

Faza	I nedelja					II nedelja					III nedelja					IV nedelja					V nedelja					VI nedelja					VII nedelja				
	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P					
Dizajniranje																																			
Implementacija																																			
Pripr. testova																																			
Testiranje																																			
Integracija																																			
Upravljanje pr.	X					X					X					X					X					X									

● Ukupno trajanje projekta: 35 dana

● Šta ukoliko faza integracije ne obuhvata integraciono testiranje, već integraciono testiranje uključuje faza testiranja?

Na primer:

za integraciju svakog slučaja korišćenja je potrebno još po 0.5 čovek/dana

# Zadatak 1 - Rešenje (b)

## ● Inkrementalni model

Faza	I nedelja					II nedelja					III nedelja					IV nedelja					V nedelja					
	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	
Dizajniranje																										
Implementacija																										
Pripr. testova																										
Testiranje																										
Integracija																										
Upravljanje pr.	X					X					X					X					X					

## ● Svaki slučaj korišćenja označen je drugom bojom.

Faza	I nedelja					II nedelja					III nedelja					IV nedelja					V nedelja					
	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	P	U	S	Č	P	
Dizajniranje	3	3	2	2	2	2	2	2	1	1	1	1	1	1	2	2	2	2								
Implementacija			1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	2	2	2	2	2	2		
Pripr. testova							1	1		1					1				1							
Testiranje									1			1				1				1				1		
Integracija										1		1				1				1				1		
Upravljanje pr.	X					X					X					X					X					

● Ukupno trajanje projekta: 23.5 dana

● Da li može efikasnije (brže)?

# Zadatak 2

- Postavljeni ste za menadžera projekta za razvoj jednog malog softverskog proizvoda. Početne procene opterećenja nalaze se u tabeli. U timu imate troje ljudi na raspolaganju, koji su angažovani sa punim radnim vremenom (full-time), svi dovoljno dobri da obavljaju bilo kakvu aktivnost na projektu.

Aktivnost	Obim posla (čovjek/dana)
Analiza zahteva	10
Dizajn sistema	12
Detaljni dizajn sistema	10
Implementacija	10
Testiranje i ispravka bagova	6
Instalacija	1



# Zadatak 2



- a) Opisati projektni plan rečima i objasniti koje sve mogu biti aktivnosti upravljanja projektom.
- b) Napraviti detaljan projektni plan i nacrtati Gantt dijagram, koji deli ove aktivnosti između tri člana tima i obuhvata prekretnice. U tabeli nisu date aktivnosti upravljanja projektom (sastanci i slično), ali potrebno je da i njih isplanirate.
- c) Koji su potencijalni rizici u vašem planu?

# Zadatak 2 - Rešenje



- Rešenje bazirano na modelu vodopada sa nekim modifikacijama
- Prepostavka je da jedna aktivnost prati drugu, ali između dve aktivnosti neki članovi mogu da počnu da rade sa novim aktivnostima
- Mali projekat
- Trajanje projekta:  
 $10+12+10+10+6+1 = 49$  dana / 3 člana tima = 16.3 radnih dana
- Upravljanje projektom - koliko traje?

# Zadatak 2 - Rešenje (a)

- Najmanje jednočasovni sastanak nedeljno, u kome učestvuju svi članovi tima
- Trajanje aktivnosti:  
 $1^h \times 3 \text{ člana} \times 3 \text{ nedelje} = 9^h$
- Druga aktivnost upravljanja projektom - upravljanje konfiguracijom (koja uključuje i obuku)
- Trajanje aktivnosti:  
obuka na početku projekta  $1^h \times 3 \text{ člana} = 3^h$   
+ upravljanje konf. tokom projekta  $1^h \times 3 \text{ člana} \times 3 \text{ nedelje} = 9^h$   
 $\Sigma = 12^h$
- Upravljanje projektom ukupno:  
 $\Sigma = 9^h + 12^h = 21^h$
- Ovo ćemo zaokružiti na  $\sim 24$  sata ili 3 čovek/dana

# Zadatak 2 - Rešenje (b)

- Prva aktivnost se odvija svake nedelje i zbog vidljivosti nije uključena u dijagramu. Takođe, ni druge aktivnosti nisu uključene u dijagramu, jer su distribuirane tokom celog projekta. Jedini deo koji je prikazan je deo obuke na početku.

Dan	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Član tima																								
Osoba #1																								
Osoba #2																								
Osoba #3																								

	Analiza zahteva
	Dizajn sistema
	Detaljni dizajn sistema
	Implementacija
	Testiranje i ispravka bagova
	Instalacija

- Prekretnice (*milestones*) predstavljaju svaku promenu boje - promenu aktivnosti.

# Zadatak 2 - Rešenje (c)



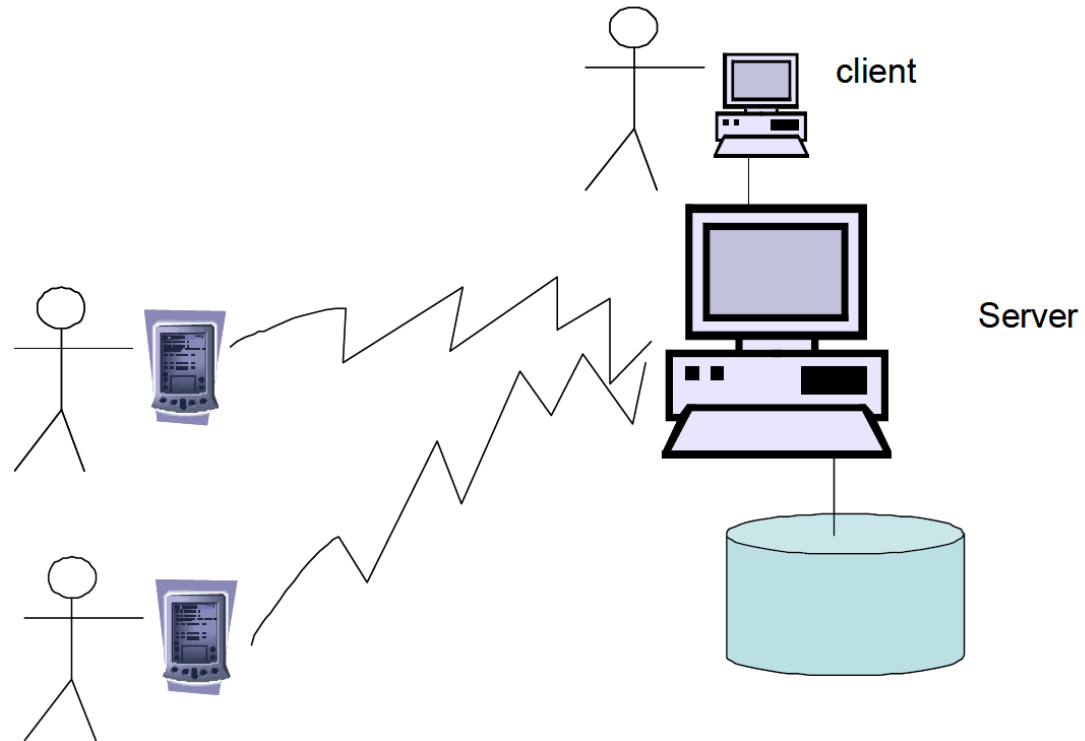
- Potencijalni rizici kašnjenja u ovom planu:
  - Ponekad jedna osoba radi samo jedan dan na kraju aktivnosti. Ovo može izazvati kašnjenje u slučaju bolesti na primer.
  - Upravljanje aktivnostima nije jasno navedeno u dijagramu, pa zbog toga može da se izazove blagi zastoј u celom projektu.
  - Jedna osoba ne može biti u stanju da instalira proizvod (zbog nedostatka znanja ili bolesti), pa isporuka može da se odloži.

# Zadatak 3



- Pretpostavimo da razvijate softver za beogradski metro. Svaki zaposleni ima svoj tablet računar na kome može da vidi sve informacije o svojim aktivnostima i aktivnostima drugih zaposlenih (ljudi na dužnosti, njihove trenutne i planirane aktivnosti, urađene aktivnosti, njihova stanja,... itd.). Svaki zaposleni može da unese/ažurira svoje aktivnosti ili ako ima dozvoljene privilegije može kreirati novu aktivnost za nekog zaposlenog. Sve informacije se čuvaju na centralizovanom serveru preko koga zaposleni komuniciraju preko svojih tablet računara (kao na slici). Sistem može da se administrira preko lokalne konekcije.

# Zadatak 3 (nastavak)



# Zadatak 3 (nastavak)



- Vi ste tim lider projektnog tima i treba da napravite projektni plan. Treba da razvijete softver za tablet PC (grafički korisnički interfejs i komunikacijski deo), softver za komunikaciju i softver potreban za server za upravljanje podacima.
- Imate tim od 6 ljudi - svi oni mogu da budu dizajneri sistema, programeri, tester ili da pišu dokumentaciju. Ugovor koji ste potpisali zahteva da implementirate kompletan softverski sistem u 25 nedelja. Sav hardver i razvojni alati su dostupni Vama i Vašem timu. U postupku rešavanja treba koristiti V model.
- a) Prepoznati glavne principe za planiranje ovog projekta (uloženi napor, vreme, opterećenje ljudi, kvalitet)
- b) Prepoznati glavne aktivnosti i podaktivnosti, i rasporediti aktivnosti među članovima projektnog tima (uključujući i Vas)
- c) Nacrtati Gantt dijagram za ovaj projekat i dijagram raspodele resursa u projektном timu



# Zadatak 3 - Rešenje (1)

## ● Glavne aktivnosti:

ID	Glavna aktivnost	Podaktivnost
A1	Celokupni sistem	
A1.1		Korisnički zahtevi
A1.2		Dizajniranje sistema
A1.3		Integracija sistema
A1.4		Validacija
A1.5		Verifikacija i ispravka bagova
A1.6		Isporuka
A2	Softver za tablet	
A2.1		Korisnički zahtevi
A2.2		Dizajniranje podsistema
A2.3		Implementacija
A2.4		Verifikacija i ispravka bagova

# Zadatak 3 - Rešenje (2)



ID	Glavna aktivnost	Podaktivnost
A3	Softver za server	
A3.1		Korisnički zahtevi
A3.2		Dizajniranje podsistema
A3.3		Implementacija
A3.4		Verifikacija i ispravka bagova
A4	Softver za komunikaciju	
A4.1		Korisnički zahtevi
A4.2		Dizajniranje podsistema
A4.3		Implementacija
A4.4		Verifikacija i ispravka bagova
A5	Upravljanje projektom	
	Obuhvata kompletan menadžment, projektnu dokumentaciju, proveru kvaliteta, tehničku podršku	

# Zadatak 3 - Rešenje - diskusija

- Opterećenje treba distribuirati na najefikasniji način
- V model se koristi u svakoj aktivnosti (A1-A4)
- Softver za komunikaciju će zahtevati manje posla, nego razvoj softvera za tablete i serversku stranu
- Alokacija resursa u timu:
  - Aktivnosti A1, A4 - inženjeri I1 i I2
  - Aktivnost A2 - inženjeri I3 i I4
  - Aktivnost A3 - inženjeri I5 i I6
- Kako se radi o V modelu, aktivnosti validacije i verifikacije dolaze sa fazom zahteva i dizajniranja sistema, a nakon implementacije i integracije.
- Inženjeri iz A2 i A3 treba da pomognu u verifikaciji/validaciji sistema.
- Stvaranje dokumentacije je deo aktivnosti u skladu sa V modelom.



# Zadatak 3 - Rešenje

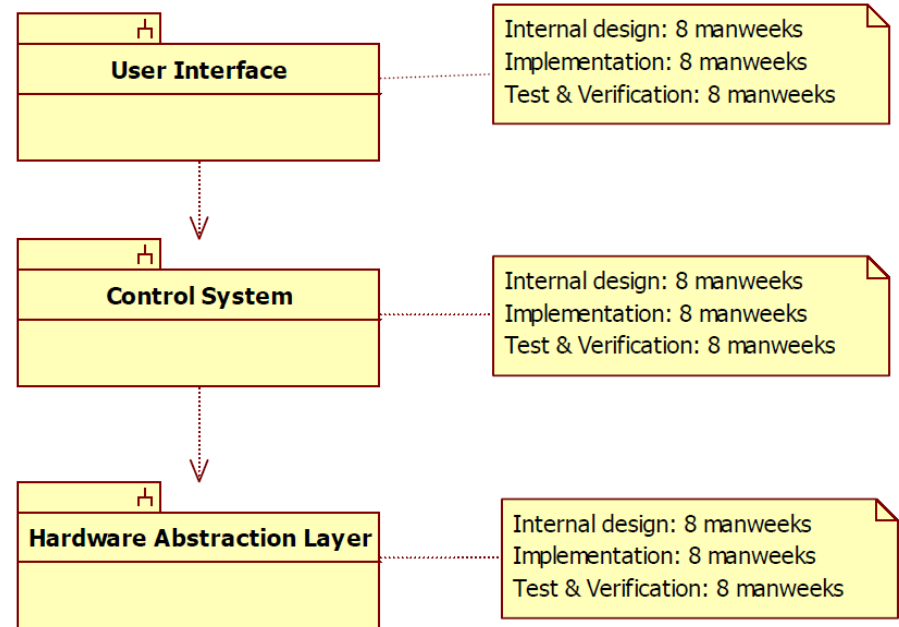
## Dijagram raspodele resursa

Inženjer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
I1	a1	a1	a1	a1	a1	a4	a4	a4	a4	a4	a4	a4	a4	a4	a4	a4	a4	a1	a1	a1	a1	a1	a1	a1	a5
I2	a1	a1	a1	a1	a1	a4	a4	a4	a4	a4	a4	a4	a4	a4	a4	a4	a4	a1	a1	a1	a1	a1	a1	a1	a5
I3	a1	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a1	a1	a1	a1	a1	a1	a5	a5
I4	a1	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a2	a1	a1	a1	a1	a1	a1	a5	a5
I5	a1	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a1	a1	a1	a1	a1	a1	a5	a5
I6	a1	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a3	a1	a1	a1	a1	a1	a1	a5	a5
Menadžer projekta	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5	a5

- Svi članovi počinju sa zahtevima i dizajniranjem sistema. I1 i I2 nastavljaju na sistemskom nivou, drugi podtimovi rade svoje podsisteme.
- Svi članovi rade verifikaciju i validaciju sistema.
- Poslednje dve nedelje su uzete za dodatni rad, ako bi bilo potrebno!

# Zadatak 4

- Radite na projektu koji će razviti softver za seriju mašina za pranje veša. Zajedno sa klijentima, definisani su svi sistemski zahtevi.



Vi ste se složili koja dugmad je potrebna i šta će biti prikazano na prednjem delu mašine za veš, i definisali ste hardverski apstrakcioni sloj, koji treba da obezbedi prenosivost za novi hardver u budućnosti (samo jedan sloj će morati da se menja ukoliko se hardver menja). Pogledati figuru.

# Zadatak 4 - nastavak



- Tim čine tri čoveka (uključujući Vas) koji rade puno radno vreme na projektu. Svako može uraditi bilo koji zadatak (aktivnost) na projektu. Do danas (6. marta) Vi i Vaš tim ste proveli 4 nedelje u dogovorima sa klijentom oko sistemskih zahteva, i možete proceniti vreme razvoja za svaki deo projekta, kao što je opisano u napomenama na slici.
- Potrebno je prikazati modele razvoja za ostatak projekta:
  - a) iterativni
  - b) inkrementalni

Za svaki od tih modela razvoja, definisati prekretnice (*milestones*) i krajnje rezultate (*deliverables*), sa datumima. Takođe ukratko opisati kako svaki model razvoja utiče na rizike i koji su mogući načini saradnje sa klijentima.

Nije neophodno crtati Gantt dijagram ili dijagram alokacije resursa, samo listu prekretnica i krajnje rezultate, kao i kratku diskusiju.

# Zadatak 4 - Rešenje



- **Pretpostavke:**

- Mi ćemo razmatrati da svi ljude uvek zajedno rade na istom zadatku
- Pretpostavljamo da će mesečno biti potrošeno 2 dana na sastanke o upravljanju projektom (ukupno 2 nedelje za ceo projekat)
- Pretpostavljamo da je dokumentacija za svaki zadatak uključena u date cifre



# Zadatak 4 - Rešenje - Iterativni model

## ● Pretpostavke:

- Svaki zadatak delimo u 2 podzadatka, npr. Implementacija 1 + Implementacija 2
- Svaki podzadatak traje 4/3 kalendarske nedelje
- Pretpostavimo da nam je potrebno 6 čovek-nedelja (tj. 2 kalendarske nedelje) za integraciono testiranje, ispravku bagova i validaciju celog sistema
- Vršimo dve iteracije, prva se završava sa isporukom osnovnih funkcionalnosti svake komponente (verzija 0.5) i poslednja se završava sa isporukom kompletnog, ispravnog sistema (verzija 1.0)

## ● Prekretnice i rezultati:

- 4. nedelja: završen Dizajn1 od UI, Dizajn1 od CS i Implementacija1 od CS
- 8. nedelja: završen Dizajn1 od HAL, Implementacija1 od UI i Testiranje1 od CS
- 13. nedelja: završena Implementacija1 i testiranje1 od HAL, i Testiranje1 od UI
- **REZULTAT: Verzija sistema 0.5 (Podzadatak1 završen za sve komponente)**
- 16. nedelja: završen Dizajn2 od UI, CS, HAL
- 21. nedelja: završena Implementacija2 od CS i HAL, i Testiranje2 od CS
- 26. nedelja: završeno Testiranje2 od HAL, Implementacija2 od UI i Testiranje2 od UI
- **REZULTAT: Verzija sistema alfa (Podzadatak2 završen za sve komponente)**
- 28. nedelja: završeno integraciono testiranje, validacija, ispravke
- **REZULTAT: Konačna verzija sistema (1.0)**

# Zadatak 4 - Rešenje - Inkrementalni mod.

## ● Pretpostavke:

- Svaki zadatak traje 8/3 kalendarskih nedelja
- Odredili smo 2 dana mesečno za sastanke oko upravljanja projektom
- Prepostaviti da postoji potrebne 2 nedelje za testiranje integracije i zato je rizik veoma veliki u ovom razvojnom modelu, pa ćemo dodati 2 nedelje za otklanjanje grešaka
- Smatramo da je dobro da se izgradi sistem od dna ka vrhu, odnosno sloj HW apstrakcije prvo, u suprotnom ćemo morati da izgradimo neke stabove, da bi se obavljali najjednostavniji testovi na gornjim slojevima

## ● Prekretnice i rezultati:

- 8. nedelja: završen HW v 1.0 (dizajn, implementacija i testiranje HW)
- **REZULTAT: HW v 1.0**
- 17. nedelja: završen CS v 1.0 (dizajn, implementacija i testiranje CS)
- **REZULTAT: CS v 1.0**
- 26. nedelja: završen UI v 1.0 (dizajn, implementacija i testiranje UI)
- **REZULTAT: Alfa verzija sistema**
- 28. nedelja: završeno integraciono testiranje, validacija, ispravke
- **REZULTAT: Konačna verzija sistema (1.0)**

# Zadatak 4 - Rešenje - Ukupno vreme

## Ukupno vreme:

- 3 komponente x 3 zadatka x 8 čovek/nedelja po zadatku / 3 čoveka  
+ 2 nedelje za upravljanje projektom + 2 nedelje za validaciju i ispravke ~ 28 nedelja

## Rezime:

- Oba modela traju približno 28 nedelja, ali su prekretnice i rezultati koji su definisani na projektu različiti za ove modele, takođe načini komunikacije sa klijentom su različiti, što dovodi do različitih vrsta rizika projekata, pa su različiti i načini rukovanja tim izuzecima.
- U iterativnom modelu isporučujemo ceo sistem na pola projekta, koji pokazuje kupcu kompletan sistem (prototip faza: sistem bez svih funkcionalnosti i koji još uvek nije pravilno testiran), što omogućava povratne informacije (npr. kupac želi nešto da promeni).
- U inkrementalnom modelu, za slučaj da kasnimo, moramo da smanjimo karakteristike komponenti ugrađenih u kasnijim fazama (UI i eventualno CS). Drugi potencijalni rizik ovog modela je da kupac u kasnijim fazama traži da se promeni nešto u nižim slojevima.



# Principi softverskog inženjerstva

## Upravljanje projektom

# Grafičko predstavljanje rasporeda

- Prikazuju podelu projekta na poslove.
- Poslovi/zadaci ne treba da budu suviše mali. Veličina treba da je takva da mogu da se završe za nedelju ili dve
- Grafovi aktivnosti pokazuju međuzavisnosti zadataka i kritičnu putanju
- Bar grafovi prikazuju raspored u odnosu na kalendarsko vreme

Najraniji početak	Trajanje	Najraniji kraj
Labela aktivnosti, opis aktivnosti		
Najkasniji početak	Kašnjenje	Najkasniji kraj

# Model mreže aktivnosti



- Prva faza u stvaranju modela mreže je da predstavimo aktivnosti i njihove međusobne veze, kao graf.
- Pravila koja treba poštovati:
  - Mreža ima samo jedan startni čvor
  - Mreža ima samo jedan krajnji čvor
  - Čvor ima vreme trajanja, jer predstavlja neku aktivnost
  - Veze nemaju vreme trajanja, jer one povezuju aktivnosti
  - Neka aktivnost može imati preduslove (ne može početi pre nego što se završi neka druga aktivnost ili više aktivnosti)
  - Vreme teče sa leva na desno
  - Poželjno je da mreža nema petlje u grafu
  - Mreža ne sme da sadrži mrtve putanje!

# Zadatak 5

- Neka je dat projekat sa sledećim aktivnostima:

Aktivnost	Trajanje (u nedeljama)	Preduslov
A Izbor hardvera	6	
B Konfiguracija sistema	4	
C Instalacija hardvera	3	A
D Migracija podataka	4	B
E Nacrt procedura	3	B
F Izbor zaposlenih	10	
G Obuka za korisnike	3	E, F
H Instalacija i testiranje sistema	2	C, D

# Metod kritične putanje



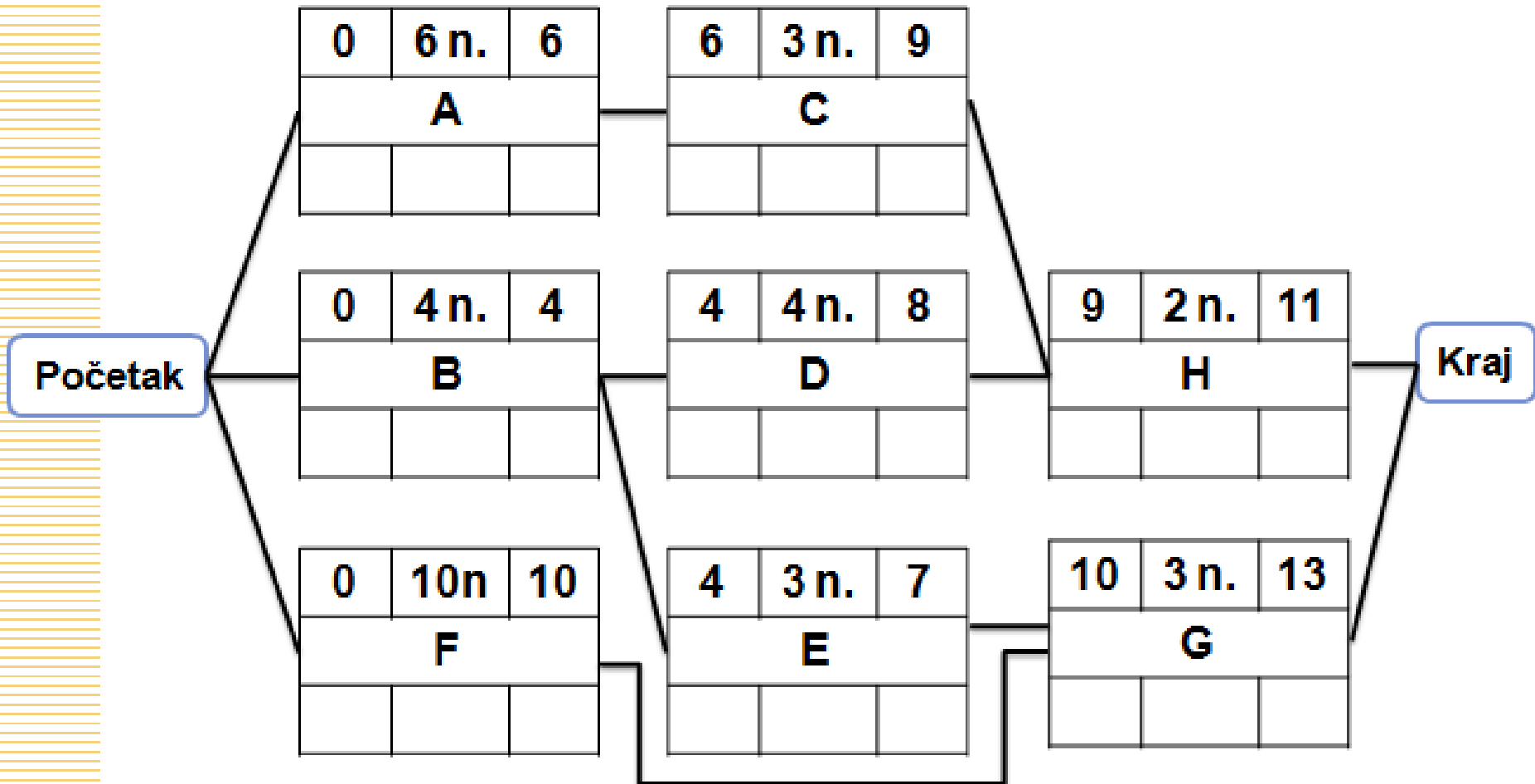
- Služi za određivanje kritične putanje u mreži aktivnosti (od koje zavisi trajanje projekta) i vremenske rezerve aktivnosti
- Dva prolaza kroz graf:
  - prolaz unapred (the forward pass)
  - prolaz unazad (the backward pass)



# Prvi prolaz CPM (unapred)

- Prolazi se kroz graf počev do startnog čvora ka završnom
- Za svaku aktivnost odredi se Early Start Time (EST) i Early Finish Time (EFT)
- EST je jednako najvećem EFT svih aktivnosti prethodnika u grafu (0 za početni čvor)
- $EFT = EST + \text{vreme trajanja aktivnosti}$

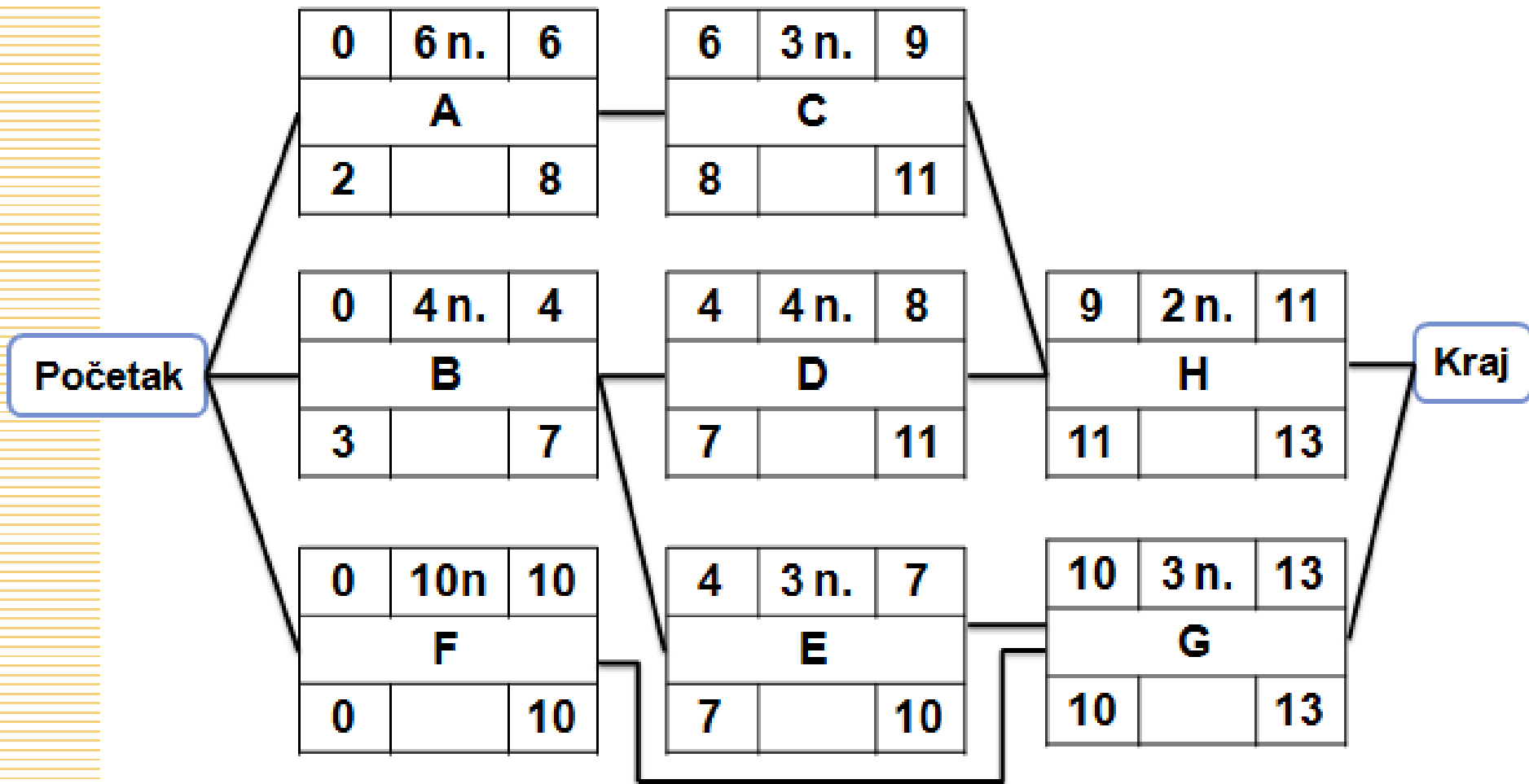
# Zadatak 5 - Rešenje (prvi prolaz)



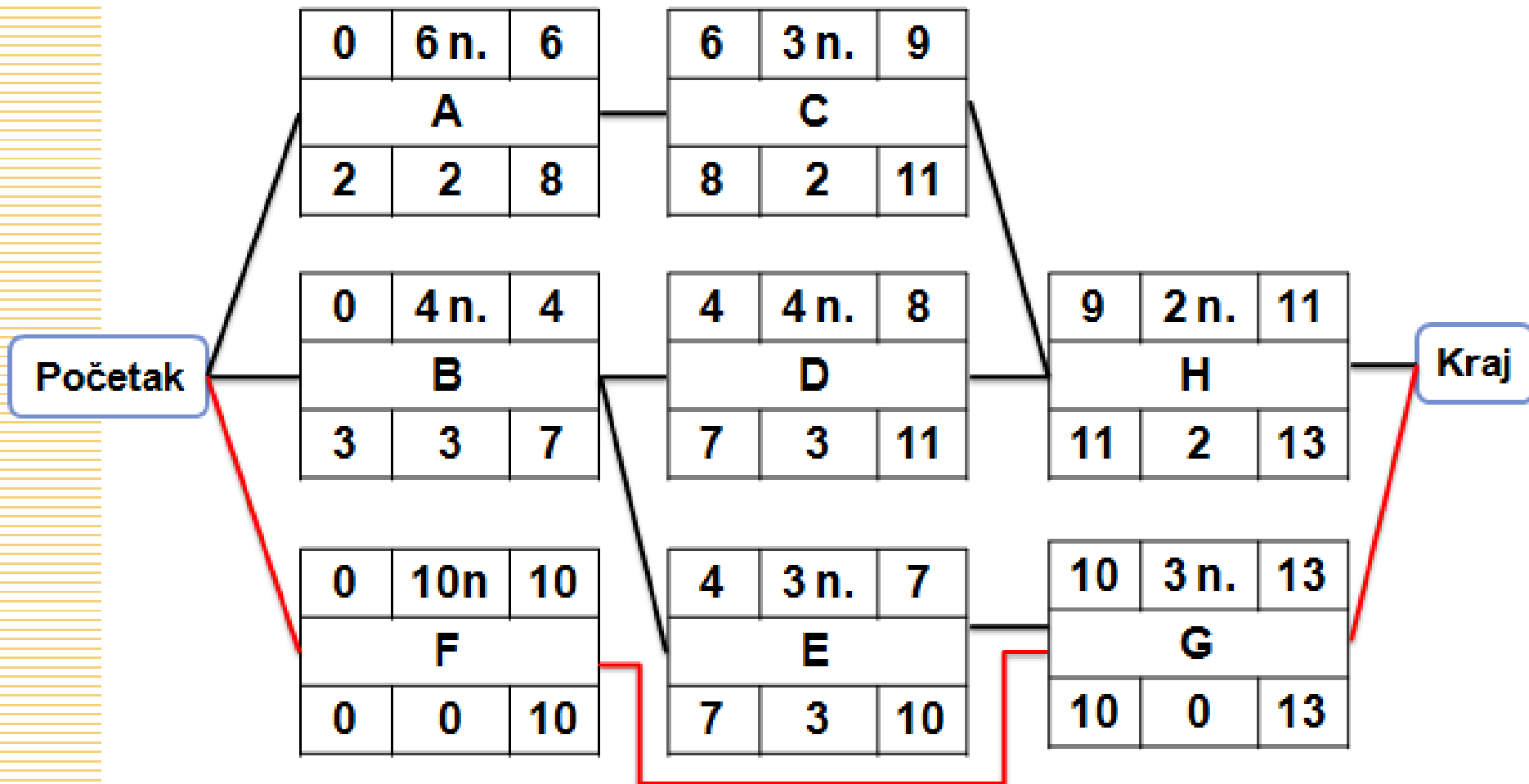
# Drugi prolaz CPM (unazad)

- Prolazi se kroz graf unazad počev do završnog čvora (čvorova) ka startnim
- Za svaku aktivnost odredi se Late Start Time (LST) i Late Finish Time (LFT)
- LFT je jednako najmanjem LST svih aktivnosti sledbenika u grafu (za završni čvor, jednako je njegovom EFT)
- $LST = LFT - \text{vreme trajanja aktivnosti}$

# Zadatak 5 - Rešenje (drugi prolaz)



# Zadatak 5 - Rešenje (kritičan put)





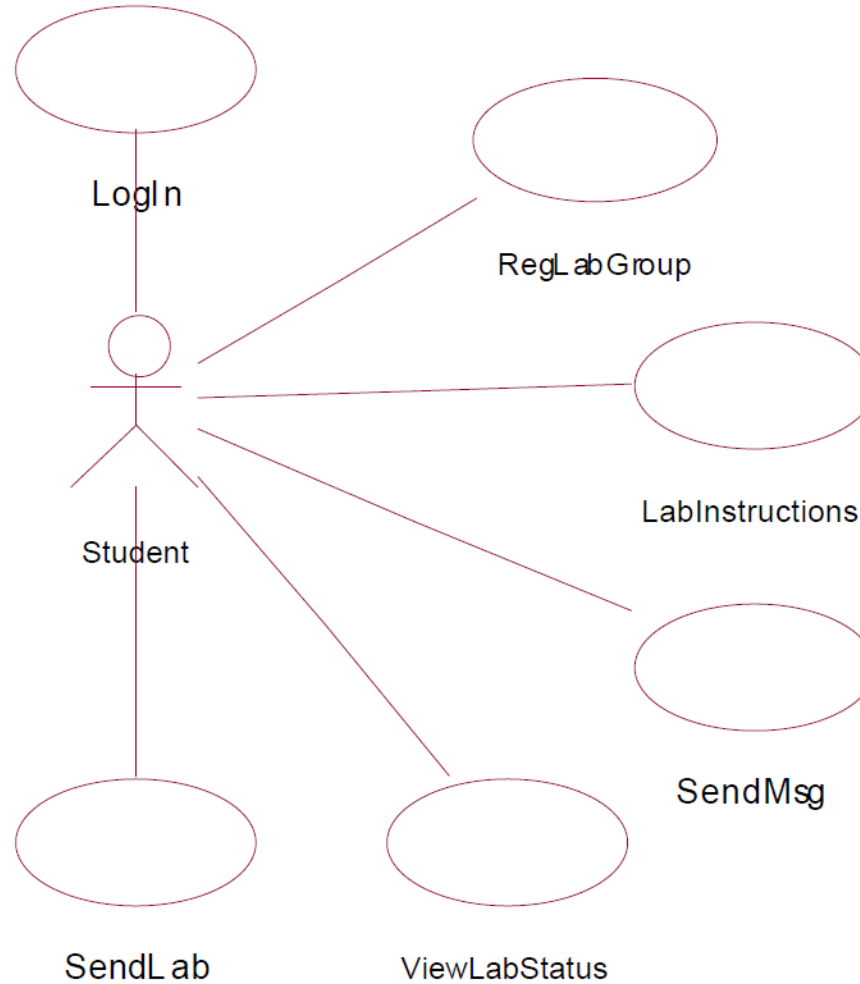
# Principi softverskog inženjerstva

## Inženjerstvo zahteva

# Zadatak 6

- Katedra za računarsku tehniku i informatiku na ETF-u realizuje softverski sistem za laboratorije. Sistem treba da omogući studentima da se prijave u grupe za lab.vežbe, pročitaju uputstva i šalju poruke asistentima u laboratoriji. Oni treba da imaju mogućnost slanja (upload) svojih domaćih zadataka, koji su odobreni i koje moraju da završe, ako su propustili rokove.
  - a) Konstruirasati UML dijagram slučajeva korišćenja (use-case) koji opisuje deo sistema, koji koriste studenti (nisu potrebne funkcionalnosti administratora i asistenta)
  - b) Za svaki slučaj korišćenja u dijagramu, napisati scenario (spisak koraka) interakcije između sistema i aktera koji učestvuju (glavni tok i moguće alternativne tokove)
  - c) Predložiti najmanje 10 zahteva ovog sistema. Nije potrebno pokriti svaki aspekt, već se fokusirati na formulisanje konkretnih i preciznih zahteva, tako da najmanje 4 bude nefunkcionalnih, i najmanje 4 funkcionalnih

# Zadatak 6 - Rešenje (a)





# Zadatak 6 - Rešenje (b)



## ● Logovanje:

- 1) Sistem prikazuje dijalog za logovanje
- 2) Student unosi korisničko ime i lozinku
- 3) Student aktivira dugme za logovanje
- 4) Sistem verifikuje lozinku
- 5) Sistem prikazuje meni sa svim predmetima, koje dati student sluša
- Alt. 4a) Sistem ne može da verifikuje lozinku - POGREŠNA LOZINKA!  
(POVRATAK NA KORAK 1)

## ● Lab\_Instrukcije:

- 1) Student bira predmet
- 2) Student bira čitanje instrukcije
- 3) Sistem prikazuje listu sa svim podržanim formatima fajlova
- 4) Student bira format
- 5) Sistem prikazuje instrukcije za slanje fajla odabranog formata

# Zadatak 6 - Rešenje (b)

- Registracija za lab. grupu:
  - 1) Student bira predmet
  - 2) Student bira opciju priključivanja grupi
  - 3) Student unosi imena studenata koji će biti sa njim u grupi
  - 4) Student potvrđuje dodavanje studenata grupi
  - 5) Sistem prikazuje sve studente nove grupe
  - 6) Student potvrđuje da je sastav grupe korektan
  - 7) Sistem izveštava da je nova grupa aktivna
  - Alt. 5a) Neki studenti su već dodati u neku drugu grupu, POVRATAK NA KORAK 3
  - Alt. 5b) Neki studenti nisu upisani na predmet, pa ne mogu da budu ubačeni u tu grupu, POVRATAK NA KORAK 3

Slično uraditi i za ostale funkcionalnosti...

# Zadatak 6 - Rešenje (c)



## ● Primeri mogućih zahteva:

- Svi korisnici moraju verifikovati svoj identitet sa kor. imenom i lozinkom (F)
- Svi predmeti koji su registrovani treba da imaju informacije o svim asistentima koji su ispitivači, o tekstu i broju domaćih zadataka (F)
- Status određenog domaćeg - poslat, nije poslat, itd. (F)
- Status roka za predaju zadataka - u toku/istekao (F)
- Kada je učenik odobren da šalje sve domaće, treba dozvoliti da može da šalje i pitanja asistentu-ispitivaču pre samog upload-a (F)
- Za sve poslate zadatke na server, ažurirati odmah listu poslatih domaćih (F)
- Sistem je veb baziran
- U sistemu treba da budu informacije o svim predmetima u tom semestru
- Student će dobijati samo informacije o predmetima na koje je registrovan
- Sistem treba da bude napisan u PHP-u