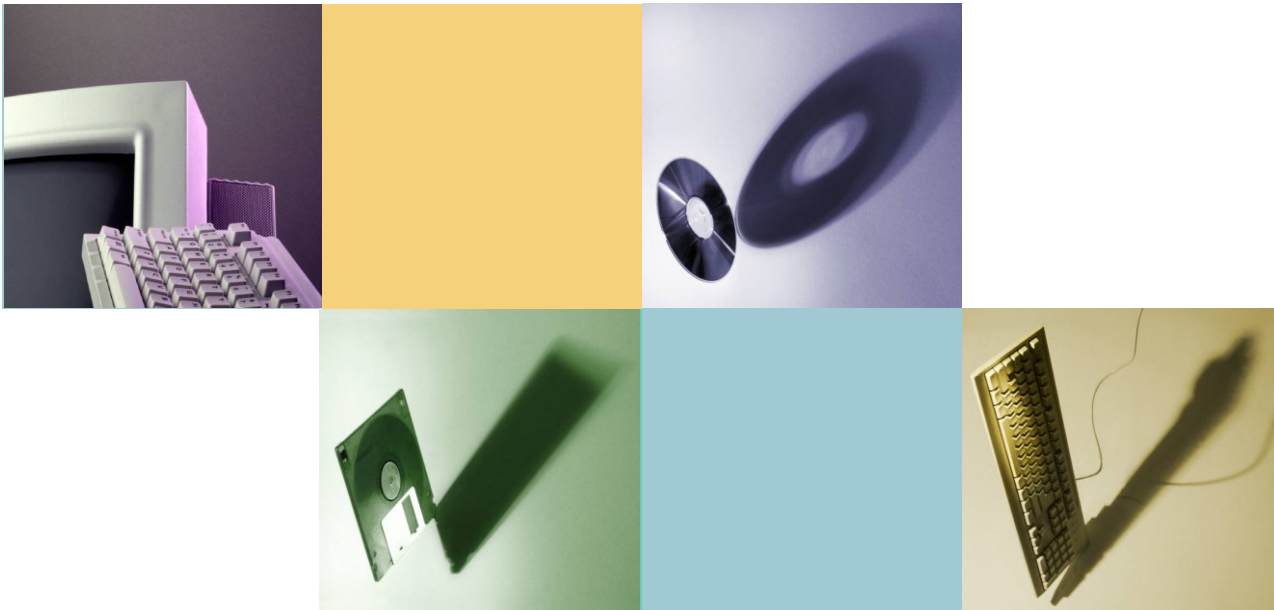


Upravljanje softverskim projektima



školska 2021/22.

Vežbe 2 i 3 (prvi blok)

Planiranje projekta - Agenda

1. Prince2 i StepWise

2. Projektne aktivnosti

3. Mreža aktivnosti – CPM i Gant

4. PERT tehnika



Prince2 i SWPMM

- **PRINCE2 (PRojects IN Controlled Environme nts)** - metod za efikasno upravljanje projektom
- PRINCE2 je grupa standarda za upravljanje projektima, koju koristi ICT britanske vlade. Ovaj standard se sada mnogo koristi i na ne-državnim projektima u Velikoj Britaniji.
- Step Wise Framework treba da bude kompatibilan sa PRINCE2.



Korak 0: Izbor projekta

- Ovaj korak je nulti, zato što izbor projekta treba da bude izvan procesa planiranja.
- Predloženi projekti se ne pojavljuju ovde, nekim drugim procesom treba odlučiti da se pokrene odabrani projekat, a ne neki drugi.
- Najčešće studija izvodljivosti (studija slučaja) za projekat sugeriše da projekat treba da ima prioritet u odnosu na neke druge.



Korak 1: Odrediti obim projekta i ciljeve

- 1.1: Identifikuje ciljeve i praktične mere za efikasno izvršavanje tih ciljeva
- 1.2: Izabrati rukovodioce projekta
- 1.3: Analiza stakeholder-a; identifikovati sve učesnike u projektu i njihove interese
- 1.4: Menjanje ciljeva u zavisnosti od analize učesnika u projektu
- 1.5: Uspostaviti metode komunikacije za sve učesnike



Korak 2: Odrediti projektnu infrastrukturu

- Obično u organizacijama postoji infrastruktura koja je dovoljna za projekat. Ako su projektni menadžeri novi u organizaciji, oni moraju biti upoznati sa infrastrukturom.
- 2.1: Odrediti vezu između projekta i strateškog planiranja
 - 2.2: Odrediti standarde i procedure
 - 2.3: Odrediti organizaciju projektnog tima



Korak 3: Analiza projektnih karakteristika

- Opšti cilj ovog koraka planiranja je da osigura da se odgovarajuće metode koriste za projekat.
- 3.1: Razlikovati projektni cilj i šta je proizvod
 - 3.2: Analizirati druge projektne karakteristike (uključujući i one zasnovane na kvalitetu)
 - 3.3: Odrediti projektne rizike višeg nivoa
 - 3.4: Uzeti u obzir zahteve korisnika koji se odnose na implementaciju
 - 3.5: Odabrati metodologiju razvoja projekta
 - 3.6: Pregled procene resursa



Korak 4: Odrediti šta će biti proizvod i aktivnosti

- 4.1: Odrediti i opisati konačni proizvod projekta
- 4.2: Dijagram toka proizvoda
- 4.3: Prepoznati instancu proizvoda
- 4.4: Napraviti idealnu mrežu aktivnosti
- 4.5: Modifikovati idealno tako da se uzmu u obzir faze i kontrolne tačke



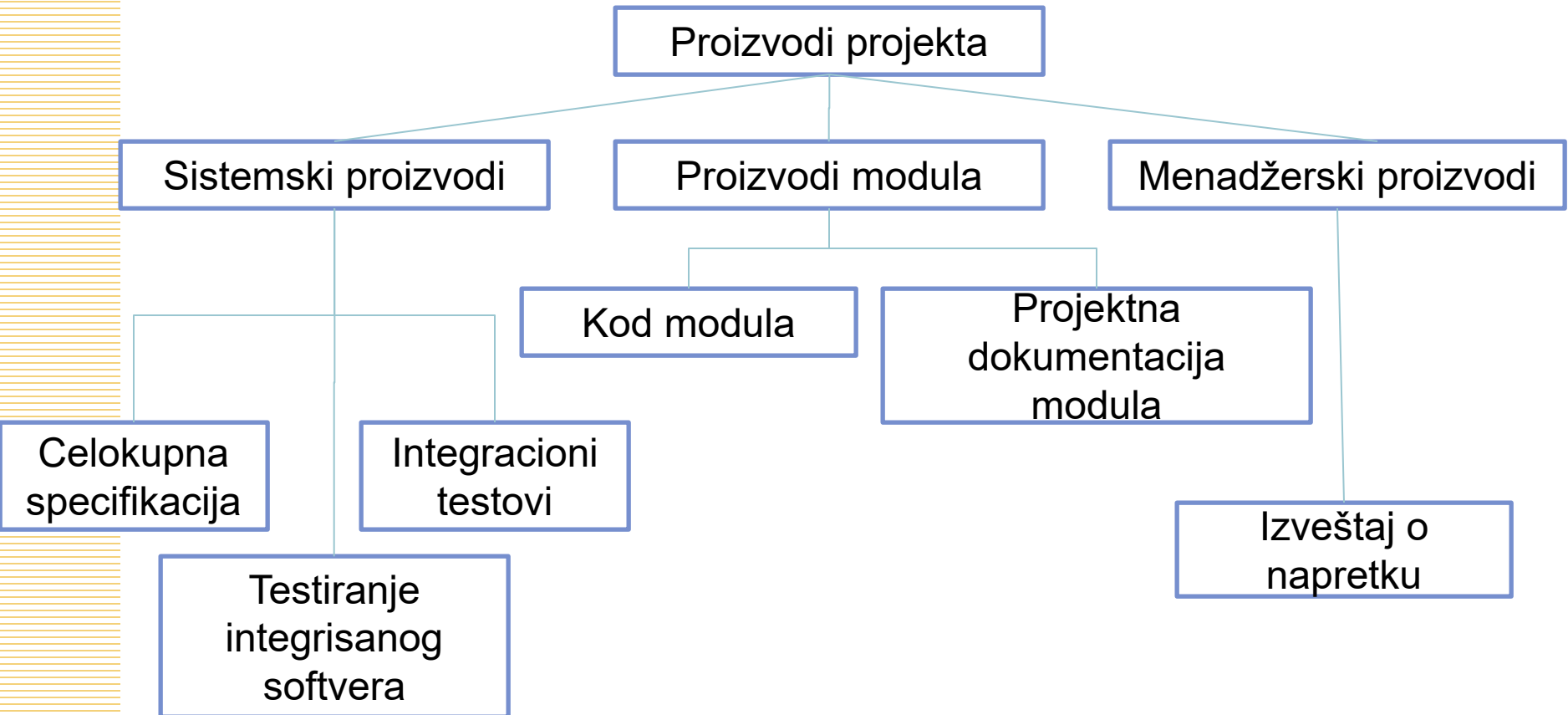
Identifikovati konačni proizvod

- Ne postoji aktivnost koja ne proizvede neki opipljiv proizvod. Neki od tih proizvoda će biti predati korisniku na kraju projekta (isporučeni proizvodi). Drugi proizvodi ne moraju biti u završnoj verziji, već služe kao pomoćni proizvodi koji se koriste u procesu formiranja isporučenog proizvoda. Ovi proizvodi obuhvataju veliki broj tehničkih proizvoda, kao što su materijali za trening i uputstva za upotrebu.
- Postoji hijerarhija, tako da glavni proizvod ima skup komponenti proizvoda, a te komponente imaju neke pod-komponente proizvoda i tako dalje... Ovi odnosi mogu biti dokumentovani kao **stablo proizvoda** - PBS (*Product Breakdown Structure*)



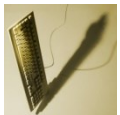
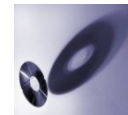
Stablo proizvoda - PBS

- *Product Breakdown Structure (PBS)*



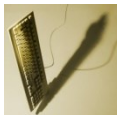
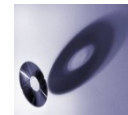
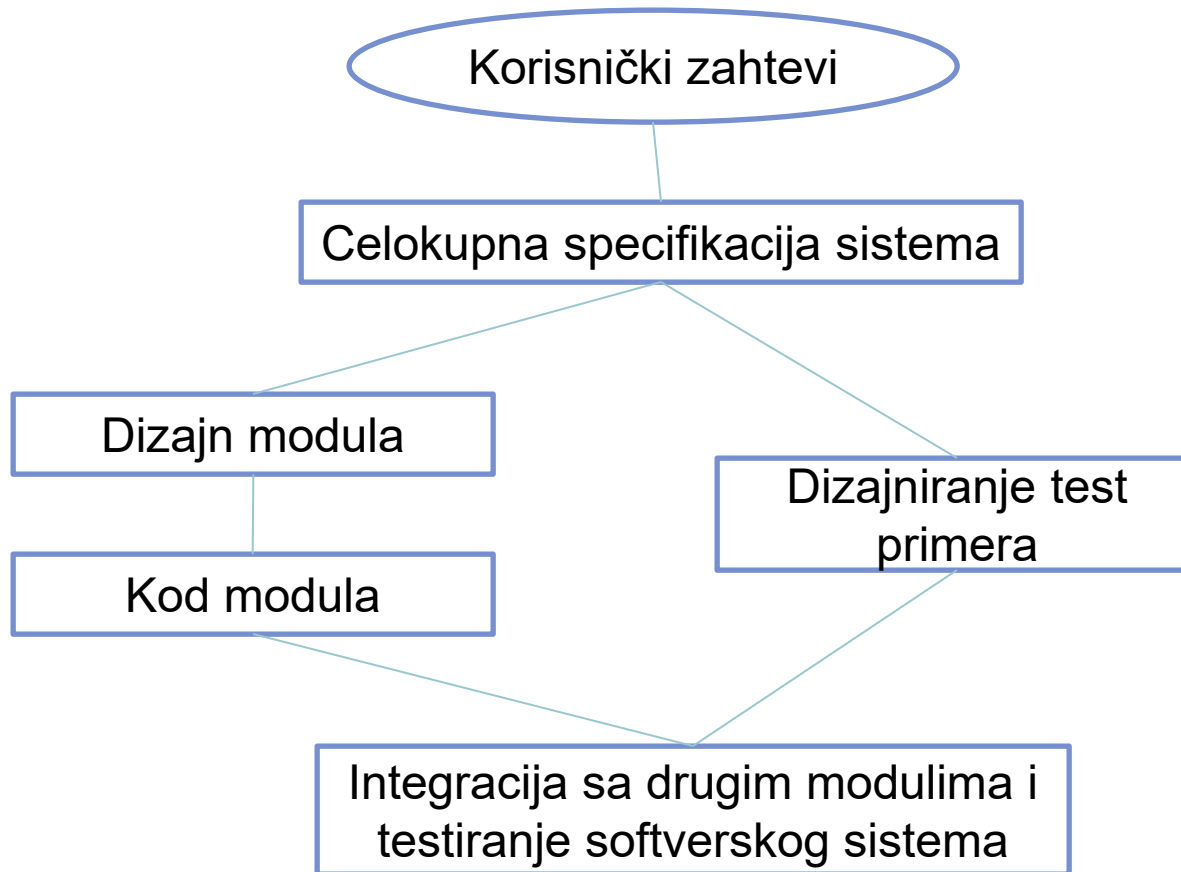
Stablo aktivnosti (rada) - WBS

- *Work Breakdown Structure (WBS)* predstavlja posao (napor) koji se ulaže i izražava se kroz aktivnosti koje se realizuju u projektu.
- *Primer (tekstualnog WBS):*
 - **33.01 MOBILIZATION AND PREPARATORY WORK**
 - **33.01.01 MOBILIZATION OF CONSTRUCTION EQUIP. AND FACILITIES**
 - **33.01.02 MOBILIZATION OF PERSONNEL**
 - **33.01.03 PRECONSTRUCTION SUBMITTALS/IMPLEMENTATION PLANS**
 - **33.01.03.01 Erosion Control Plan**
 - **33.01.03.04 Environmental Protection Plan**
 - **33.01.03.14 Construction Quality Control Plan**
 - **33.01.04 SETUP/CONSTRUCT TEMPORARY FACILITIES**
 - **33.01.05 CONSTRUCT TEMPORARY UTILITIES**
 - **33.02 MONITORING, SAMPLING, TESTING, AND ANALYSIS**
 - **33.02.03 AIR MONITORING AND SAMPLING**
 - **33.03 SITE WORK**
 - **33.03.04 ROADS/PARKING/CURBS/WALK**
 - **33.03.04.03 Aggregate Surfacing**



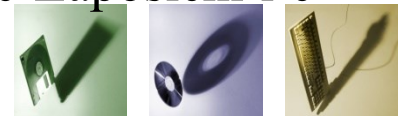
Dijagram toka proizvoda - PFD

- *Product Flow Diagram (PFD)*



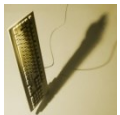
Zadatak 1: PBS i WBS (1)

- Katedra za RTI pravi svoj novi veb sistem. Projekat se sastoji iz tehničkih, netehničkih i upravljačkih poslova. Prvo se određuje cilj i piše apstrakt projekta, pravi se projektni plan, plan budžeta i plan rizika. Izrada veb sistema počinje analizom korisničkih zahteva. Korisnici sistema su studenti i nastavnici. Nakon toga pravi se prototip nove aplikacije i pišu se scenariji slučajeva upotrebe. Modelovanje aplikacije obuhvata modelovanje baze podataka, modelovanje dijagrama sekvenci i slučajeva upotrebe, izbor projektnih uzoraka koji će da se koriste u implementaciji. Implementacija se radi kroz razvoj troslojne veb arhitekture: frontend, backend i sloja podataka. Izbor tehnologija za frontend i backend treba da bude izvršen pre faze implementacije. Frontend čine izbor boja, izbor fontova i formatiranje layout-a. Backend čine funkcionalnosti za pregled i pretragu predmeta, unos novih predmeta i njihovog sadržaja, pregled i pretragu nastavnika, pregled, unos i ažuriranje vesti, komentarisanje vesti, opcije objave vesti na društvenim mrežama FB i TW. Sloj podataka treba da komunicira sa bazom podataka, ali i sa drugim sistemima: e-Student, e-Zaposleni i e-StudentskaSluzba.



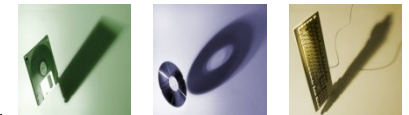
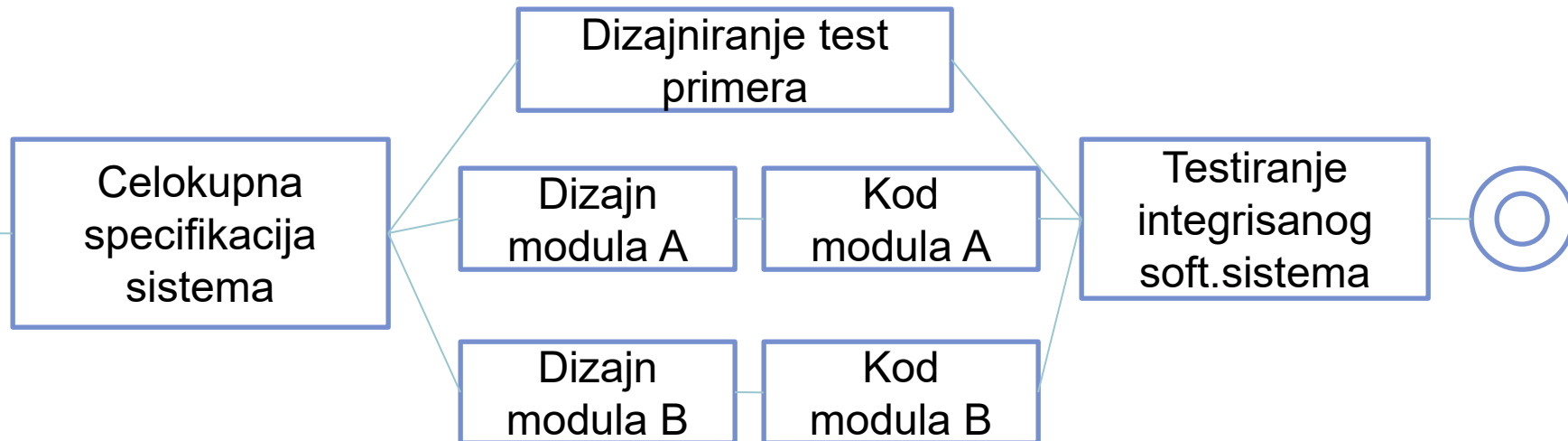
Zadatak 1: PBS i WBS (2)

- Kada se sve realizuje, potrebno je veb sistem postaviti na testno okruženje na jedan server, zatim veb sistem testirati performansno, funkcionalno i putem testiranja korisničkog interfejsa. Zatim veb sistem se stavlja na produkcionu server. Pre stavljanja na produkciju potrebno je uraditi migraciju podataka sa starog veb sajta na novi. Dva servera treba da se nabave u okviru projekta, procedurom javne nabavke, sa minimum tri ponuđača. Nakon otvaranja ponuda piše se izveštaj o nabavci. Projekat se završava finalnim izveštajem.
- **Napraviti stablo proizvoda (PBS) ovog projekta, sa izdvojenim sistemskim proizvodima, proizvodima modula i menadžerskim proizvodima.**
- **Napraviti stablo rada (WBS) ovog projekta, sa dubinom od bar 4 nivoa. Ovo stablo možete realizovati i u tekstualnom obliku.**
- Tehničke poslove, koje se odnose na razvoj softverskog proizvoda, nije potrebno dodavati, ali skup netehničkih i upravljačkih poslova možete proširiti sa još nekim proizvodima/aktivnostima.



Mreža aktivnosti

- Identifikujući aktivnosti u projektu, možemo da stvorimo mrežu aktivnosti, koja pokazuje zadatke koji moraju da se sprovode i redosled kojim ti zadaci moraju biti izvršeni.



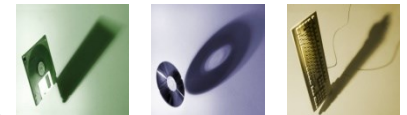
Korak 5: Proceniti rad za svaku aktivnost

5.1: Obaviti procene, od dna na gore

Procene (rada zaposlenih, proteklog vremena, ne-kadrovskih resursa) zavise od vrste aktivnosti.

Treba obratiti pažnju na razliku između proteklog vremena i rada! Rad je neki posao koji treba da se završi. Proteklo vreme je vreme između početka i kraja zadatka.

Da bismo dobili ukupnu procenu, od dna na gore, potrebno je da saberemo sve procene za individualne aktivnosti.



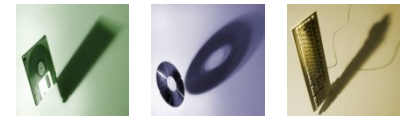
Korak 5: Proceniti rad za svaku aktivnost

5.2: Izmeniti plan da bi kontrolisali aktivnosti

Procene za individualne aktivnosti mogu se izračunati, ali neke aktivnosti traju predugo. Projekat koji ima duge aktivnosti, teško je kontrolisati.

Na primer, ako aktivnost uključuje testiranje u trajanju od 12 nedelja, teško je reći posle 6 nedelja da je završeno 50% posla. Zato je bolje izdeliti testiranje u nekoliko serija.

Ako se izveštaji koriste za monitoring i kontrolisanje, pokušajte da i aktivnosti traju koliko i period tih izveštaja.



Korak 6: Odrediti rizike

6.1: Identifikuje i broji rizike zasnovane na aktivnostima

6.2: Plan smanjenja rizika i nepredviđenih mere, gde je to moguće

Plan nepredviđenih situacija (rezervni plan) treba preduzeti ako se rizik ostvari.

6.3: Prilagoditi ukupne planove i procene, da uzmu u obzir određene rizike

Možemo dodati novu aktivnost koja će redukovati rizik.



Korak 7: Raspodela resursa

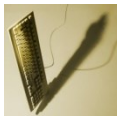
7.1: Utvrditi i rasporediti resurse

Utvrditi ko je od osoblja na raspolaganju za projekat, i privremeno im dodeliti neke zadatke.

7.2: Izmeniti planove i procene tako da imate u vidu ograničenje resursa

Neki zaposleni mogu biti potrebni istovremeno na više zadataka i u tom slučaju se gleda koji je zadatak/projekat prioritetniji. Odluke u ovom slučaju mogu imati uticaj na ukupno trajanje projekta, jer su neki poslovi odloženi, dok se čeka da zaposleni postanu slobodni.

Proizvod ovog koraka je Gantt dijagram!



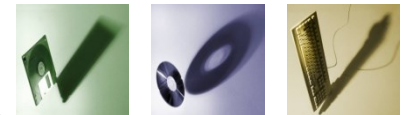
Korak 8: Pregled / objava plana

8.1: Razmatranje kvaliteta

Svaki zadatak ima kriterijum kvaliteta i zadatak može da bude završen, tek kada prođe proveru kvaliteta.

8.2: Dokumentovati plan i dobiti saglasnost

Važno je da se plan pažljivo dokumentuje, da se svi učesnici u projektu razumeju i da se slažu u obavezama koje se traže u planu.



Koraci 9 i 10: Izvršavanje plana i niži nivo planiranja

- Kada je projekat u toku, planovi će morati da budu sa više detalja, za svaku aktivnost kako dospeva.
- Detaljno planiranje kasnijih faza biće odloženo zbog toga što više informacija postaje dostupno kako se bližimo početku tih faza.
- Ali neophodno je napraviti makar privremene planove za dalje zadatke, zato što kada planiramo šta treba raditi, možemo doći do nekih potencijalnih problema.



Izbor metodologija i tehnologija (1)

- U kontekstu razvoja ICT sistema i softverskom inženjerstvu, termin metodologija opisuje grupu metoda. Metod posmatramo kao opšti način izvršavanja nekog zadatka, koji bi se mogao primeniti na bilo koji projekat, na kome je potrebno da se izvršiti isti taj zadatak.
- Tehnike imaju tendenciju da uključe primenu naučnih, matematičkih ili logičkih principa za rešavanje određene vrste problema.
- Metod često uključuje stvaranje modela. Model je reprezentacija sistema koja uvodi neke funkcije, ali zanemaruje druge.



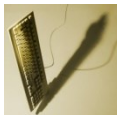
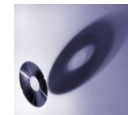
Izbor metodologija i tehnologija (2)

- Izbor metodologija i tehnologija će uticati na:
 - trening za zaposlene na razvoju
 - tip zaposlenih koji će biti angažovan
 - razvojno okruženje - zajedno i hardver i softver



Analiza projektnih karakteristika

- Da li će biti implementiran sistem orijentisan ka podacima ili sistem orijentisan ka dešavanjima?
- Da li će softver koji proizvodimo biti opšteg tipa ili specifičan?
- Druge specifičnosti softvera (konkurentno programiranje, sistem zasnovan na znanju, korišćenje računarske grafike,...)
- Da li je sistem dizajniran primarno za obavljanje unapred definisane servise ili da bude zanimljiv i zabavan?
- Kakvo je hardversko-softversko okruženje u kome sistem treba da funkcioniše?

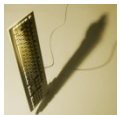
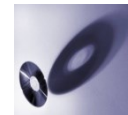


PLANIRANJE AKTIVNOSTI



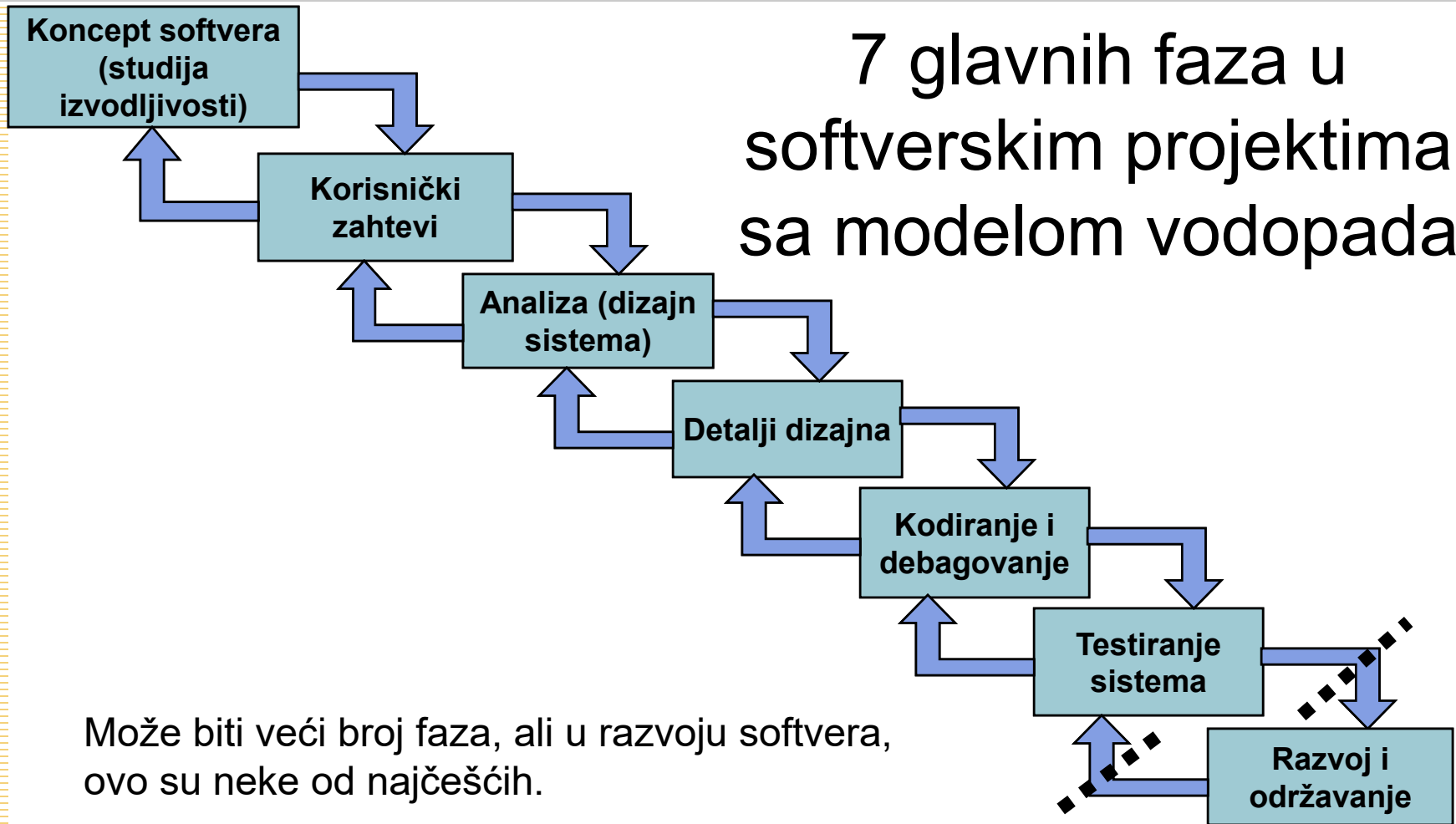
Organizacija projektnih aktivnosti

- Aktivnosti u projektu treba da su organizovane da proizvedu vidljive rezultate da rukovodstvo može da proceni napredak.
- *Milestones* (referentni datumi) su završne tačke projektnih aktivnosti.
- *Deliverables* (proizvodi za isporuku) su rezultati projekta za isporuku kupcima.
- Kod procesnog modela vodopada definicija prekretnica je pravolinijska (krajevi pojedinih faza).
- Planiranje zavisi od modela životnog ciklusa:
 - Model vodopada (eng. waterfall)
 - Spiralni model
 - Inkrementalni model
 - Agilne metodologije (XP, Lean, Kanban, Scrum)

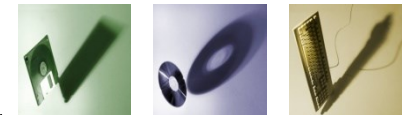


Model vodopada

7 glavnih faza u softverskim projektima sa modelom vodopada



Povratne grane – nema ih u osnovnom modelu vodopada, ali ima i modifikovanom.



Faza: Korisnički zahtevi

- Korisničke zahteve daju naručiocu posla (klijenti) u fazi analize korisničkih zahteva.
- Šta potencijalni korisnici i njihovi menadžeri zahtevaju od novog sistema.
- To može da se odnosi na funkcionalnosti koje novi sistem treba da pruži.
- A može da bude npr. i zahtev za kvalitetom (koliko dobro funkcionalnosti treba da rade)



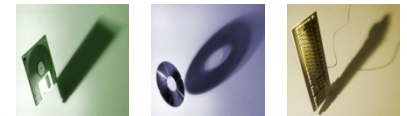
Faza: Dizajn arhitekture sistema

- Identifikovati komponente sistema
- Postojeće komponente
- Nove komponente
- Da li komponenta mora biti samo softver?
- Često dizajn arhitekture sistema može biti iz više faza:
 - dizajn celokupnog sistema
 - detaljniji dizajn
- Detaljniji dizajn = svaku soft. komponentu čini određeni broj softverskih jedinica, koje se mogu odvojeno kodirati i testirati.
Dizajn tih jedinica se radi posebno.



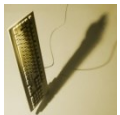
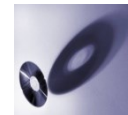
Faze: Kodiranje i testiranje

- Kodiranje i testiranje = odnosi se na svaku softversku jedinicu, pisanje koda i pronalaženje nedostataka
- Integracija
- Kvalifikaciono testiranje = sistem se pažljivo testira kako bi se osiguralo da su svi uslovi ispunjeni



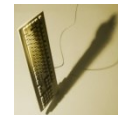
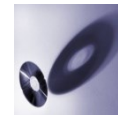
Faze: Instalacija i podrška

- Instalacija je proces osposobljavanja novog sistema i puštanje u rad.
- Šta obuhvata instalacija?
- Šta obuhvata podrška?
- Termini:
 - softver u produkciji (na produkcionom serveru)
 - nešto novo u softveru (*update*)
 - neko novo poboljšanje u softveru (*upgrade*)
 - nešto novo (*update*) nije uvek i poboljšanje!
 - softverska zakrpa (*patch, fix*)



Spiralni model

- Procesni model koji se vodi rizicima
- Obuhvata elemente jednog ili više različitih modela, kao što su inkrementalni, model vodopada, ili evolutivni korišćenjem prototipa



Prototip softvera (1)

- Prototip predstavlja nepotpunu verziju softvera, koja ima za cilj da prikaže funkcionalnosti koje će biti razvijane u softveru.
- Neki od razloga za prototipom:
 - Učite radeći (na greškama se uči!)
 - Poboljšana komunikacija
 - Poboljšano korisničko učešće u projektu
 - Pojasniti delimično poznate zahteve
 - Demonstriranje kompletnosti specifikacije



Prototip softvera (2)

- Veliki problem kod prototipa je kontrolisanje promena za sugestije koje dobijamo od strane klijenata i krajnjih korisnika.
- Sve promene možemo podeliti u 3 kategorije:
- a) kozmetičke (~ 35% promena), koje su:
 - izvršene i zapisane
- b) lokalne (~ 60% promena)
 - izvršene, zapisane, kopirane (da mogu da se uklone u kasnijim fazama), pregledane retrospektivno
- c) globalne (~ 5% promena)
 - promene koje utiču na više od jednog dela sistema



Model inkrementalne isporuke

- Sistem treba podeliti na manje komponente tzv. module, koji se u nizu implementiraju i isporučuju korisnicima. Svaki modul koji isporučimo mora da prikaže neku korist za korisnika.
- Koje su prednosti, a koje mane ovog pristupa?



Plan inkrementalne isporuke modula

- Priroda svakog modula i redosled kojim će moduli biti isporučeni klijentu, treba da budu isplanirani unapred.
- Ovaj proces je sličan strateškom planiranju, ali na detaljnijem nivou. Pažnja je više posvećena modulima koji se isporučuju korisnicima, a manje celoj aplikaciji.
- Elementi plana su: ciljevi celog sistema, inkrementalni plan i plan otvorenih tehnologija.



Plan otvorenih tehnologija

- Ako sistem treba da bude takav da stalno mogu da se dodaju nove komponente/moduli, on mora da bude proširiv, prenosiv i održiv.
- U tom slučaju treba koristiti:
 - standardni jezik višeg nivoa
 - standardne operativne sisteme
 - male module
 - promenljive parametre (podaci da mogu da se zamene bez intervencije programera)
 - standardni sistem za upravljanje bazom podataka



Inkrementalni plan

- Primer: novi sistem koji zamenjuje stari sistem i prvi moduli treba da budu delovi starog sistema (koristimo podatke iz stare baze podataka i pravimo neki GUI)
- Odnos Vrednost-Cena se koristi da se odredi redosled razvoja modula:

$$ratio = \frac{V}{C}$$

- Na primer V može imati vrednost 1-10 koja prikazuje vrednost za korisnika, a C sa vrednošću 0-10 predstavlja cenu (troškove). Što je veći ovaj odnos, prioritet je veći.
- Da li cena (troškovi) mogu biti 0?

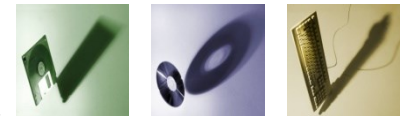
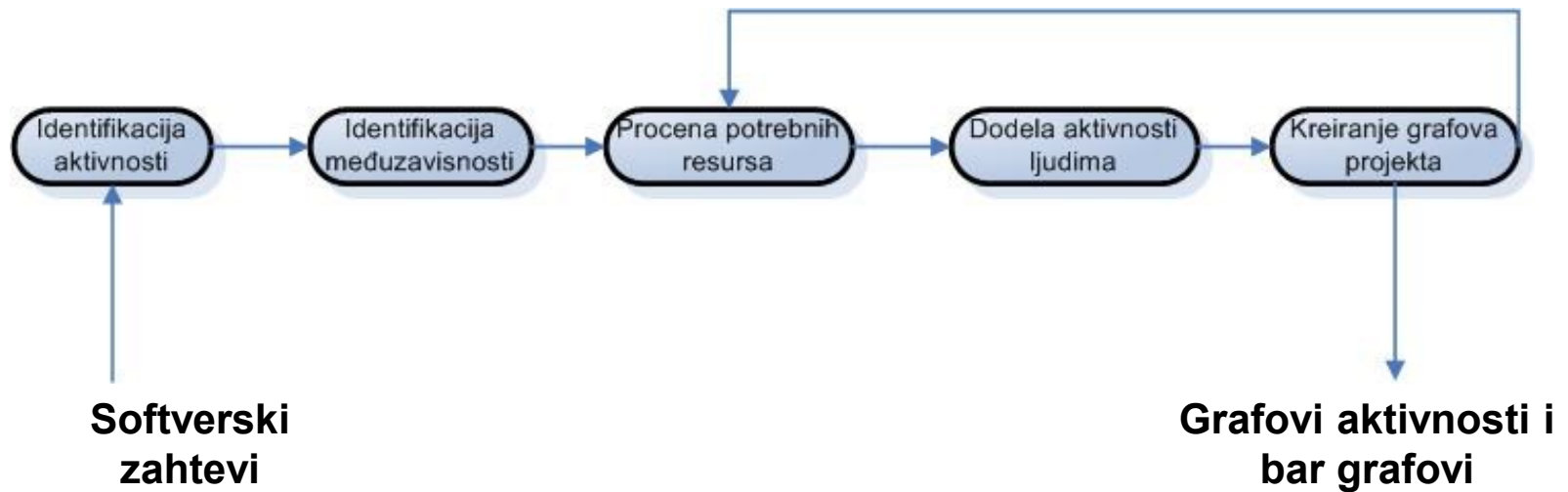


Raspoređivanje projektnih aktivnosti

- Izdeliti projekat u zadatke/poslove (eng. *tasks*) i proceniti potrebno vreme i resurse za završetak svakog od poslova
- Organizovati poslove u paraleli zbog optimalne upotrebe radne snage
- Minimizovati međuzavisnosti poslova da bi se izbeglo kašnjenje usled čekanja jednog posla na završetak drugog
- Prethodno navedeno zavisi od iskustva i osećaja rukovodioca projekta

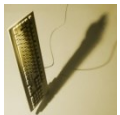
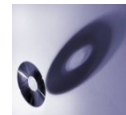


Proces raspoređivanja aktivnosti



Problemi u raspoređivanju aktivnosti

- Teško je proceniti težinu problema i sa tim povezan trošak dolaska do rešenja.
- Produktivnost nije proporcionalna broju ljudi koji rade na nekom poslu.
- Dodavanje ljudi na projekat koji kasni ima efekat još većeg kašnjenja zbog dodatnog vremena koje se troši na komunikaciju.
- Neočekivane stvari uvek se mogu desiti. U planiranju se uvek moraju predvideti vanredne situacije.



Grafičko predstavljanje rasporeda aktivnosti

- Prikazuju razdelu 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



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!



Labele u mreži aktivnosti

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



Zadatak 2: Mreža aktivnosti

Za dati projekat koji se sastoji iz 8 aktivnosti (označene oznakama A-H), dato je trajanje svake aktivnosti u nedeljama, i zavisnost (preduslov). Tamo gde nije označeno da postoji preduslov, ta aktivnost je nema.

Realizovati mrežu aktivnosti za dati projekat i odrediti kritičan put.

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 (*Critical Path method*)

- 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*)

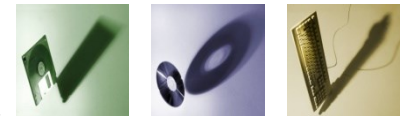
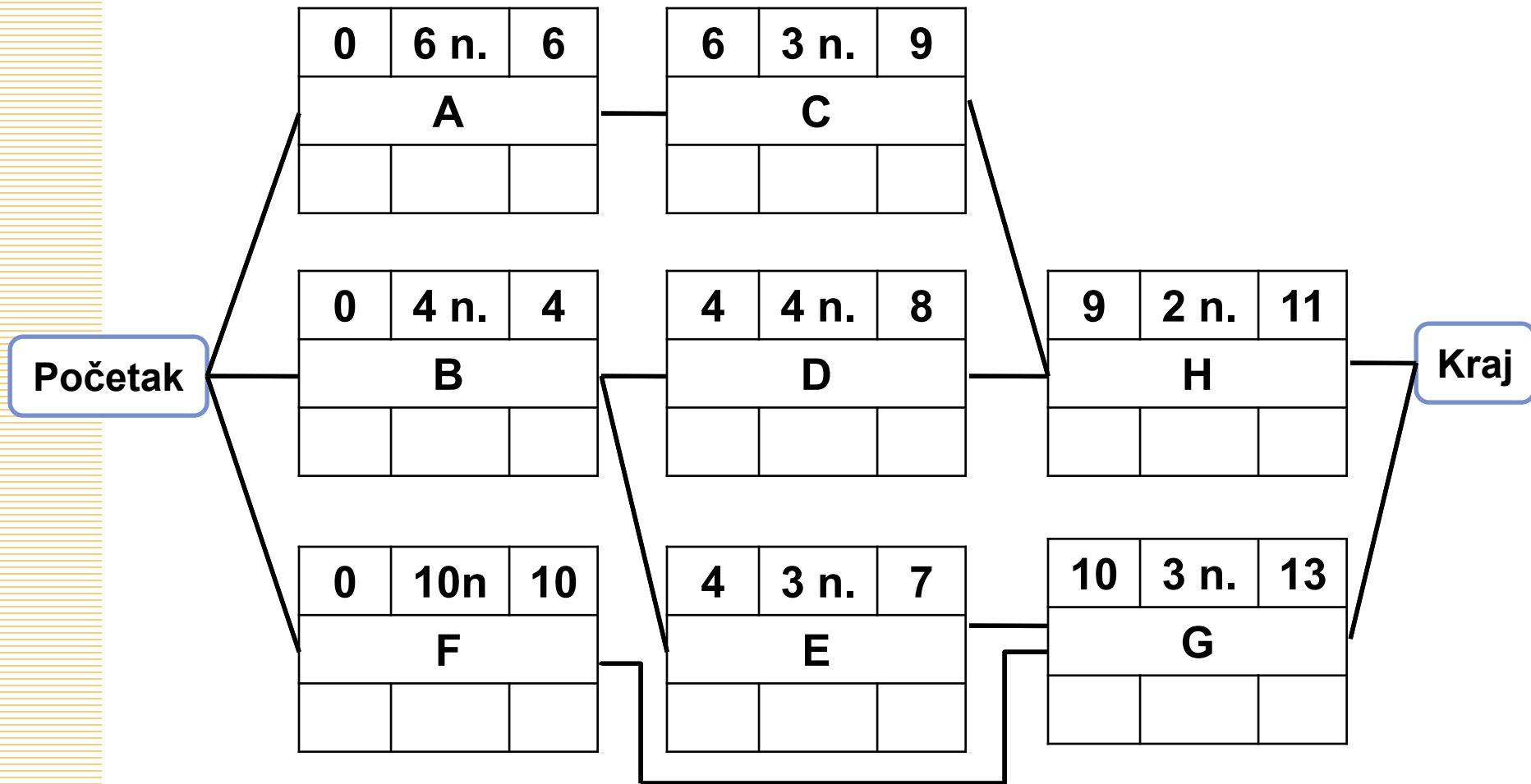


Prvi prolaz CPM

- 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)
- $EFT = EST + \text{vreme trajanja aktivnosti}$



Prvi prolaz - prolaz unapred

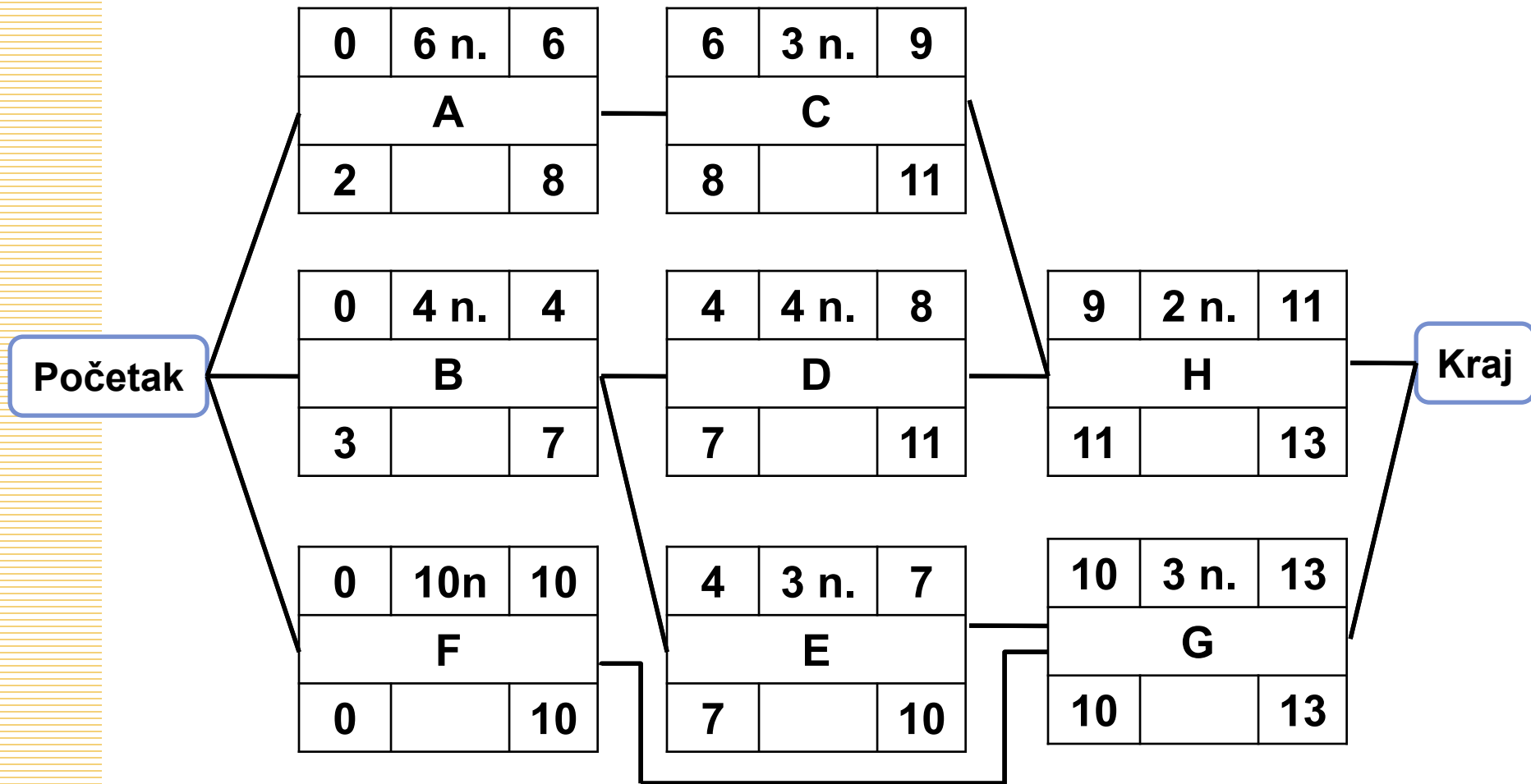


Drugi prolaz CPM

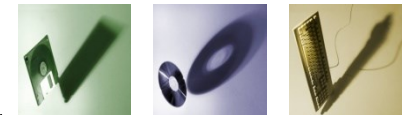
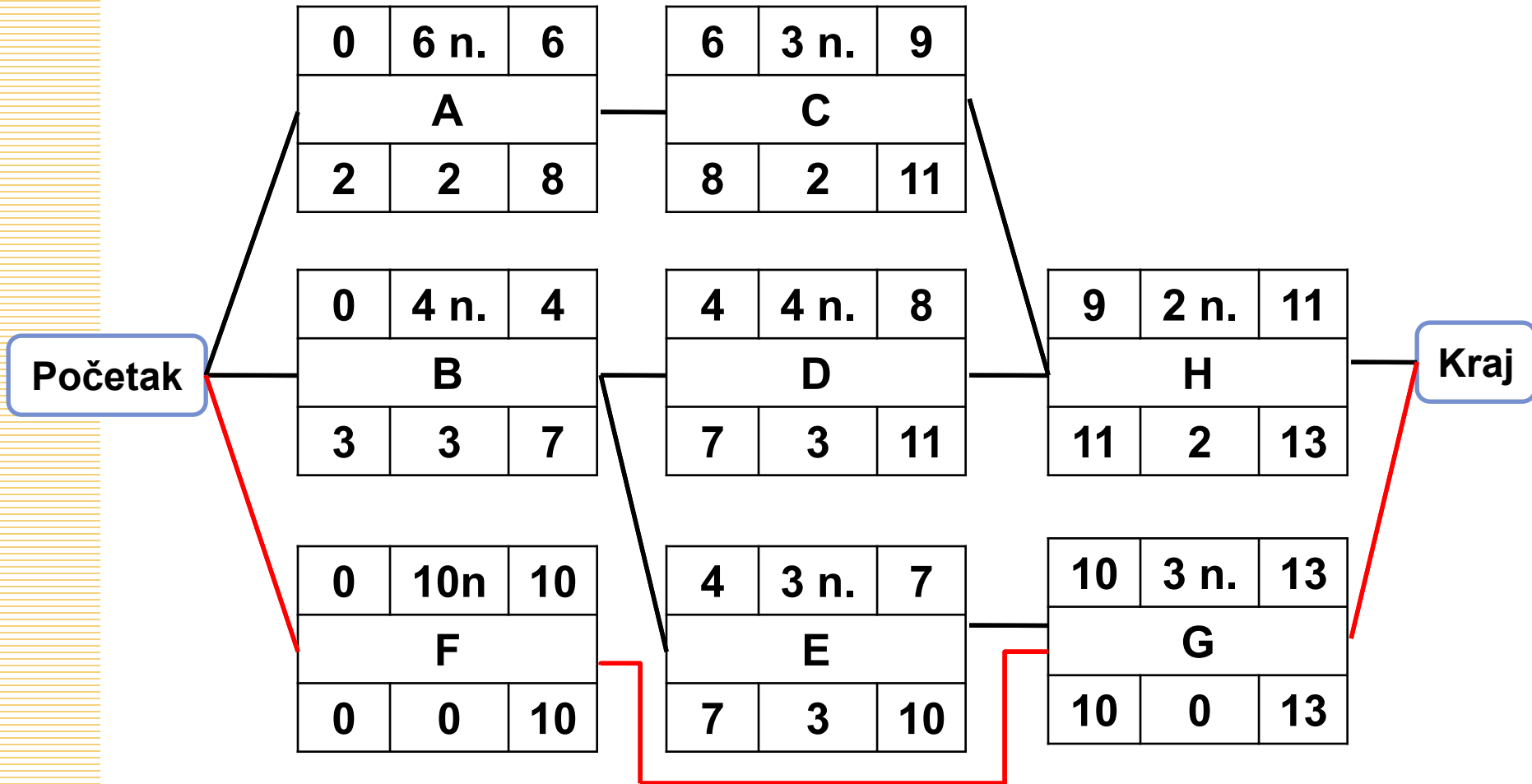
- Prolazi se kroz graf unazad počev do završnog čvora (čvorova) ka startnim
- Za svaku aktivnost odredi se Late Start Time (LST) i Late Finish Time (LFT)
- LFT je jednako najmanjem LST svih aktivnosti sledbenika u grafu (za završni čvor, jednako je njegovom EFT)
- $LST = LFT - \text{vreme trajanja aktivnosti}$



Drugi prolaz - prolaz unazad

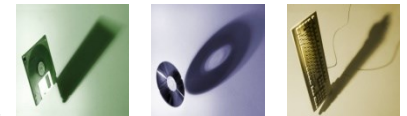


Kritični put



Kako se računa kritični put?

- Vremenska rezerva aktivnosti = $LFT - EFT$
- Aktivnosti na kritičnoj putanji po definiciji imaju vremensku rezervu 0
- Na sledećem dijagramu prikazana je vremenska linija aktivnosti za neku mrežu aktivnosti (zasenčeni deo je vremenska rezerva)



Aktivnosti u planu

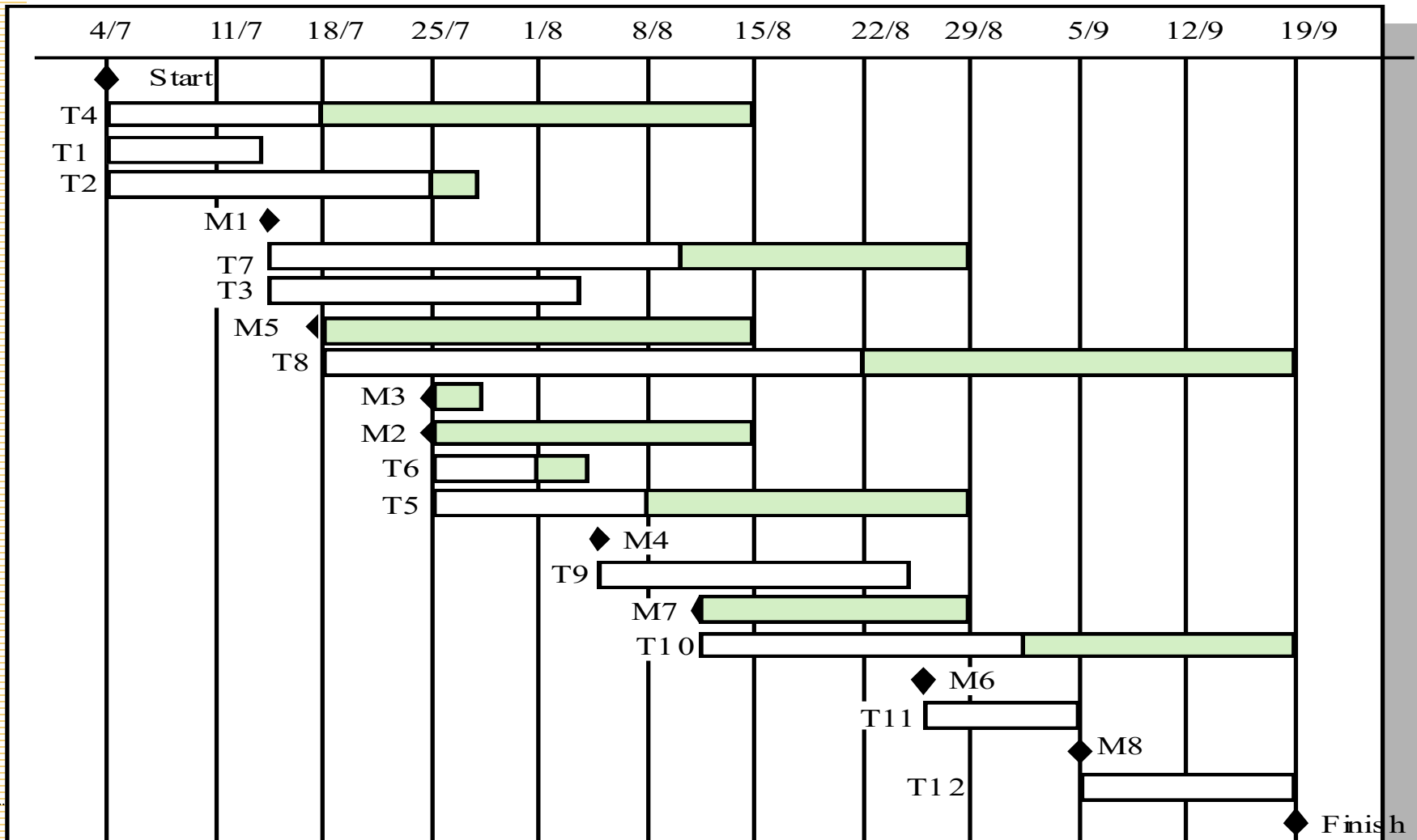
- datum početka i kraja aktivnosti
- ko će sprovesti tu aktivnost
- šta je potrebno od alata i informacija

Najčešće je izlaz jedne aktivnosti,
ulaz u drugu aktivnost!



Vremenska linija aktivnosti

Pokazuje u kalendarskom vremenu projekta, koja aktivnost se kada radi.
Vremenska rezerva (moguće kašnjenje) se obično „zasenči“ (zeleno na slici).
Poželjno obeležiti i dane koji su prekretnice (oznakama m1-m8 na slici).



GANTT

- Vremenska linija aktivnosti naziva se i **Gantt dijagram**, prema istraživaču Henriju Gantu.
- Zove se i **harmonogram**.



Zadatak 3: Gantt dijagram

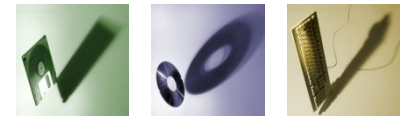
Neka kompanija realizuje svoj novi softverski proizvod zasnovan na modernim klijent-server veb tehnologijama. Funkcionalnosti koje softver treba da sadrži opisane su u 6 slučajeva korišćenja. Projektni tim ulaže napor, koji je izražen u jedinici čovek/dan i dat je sledećom tabelom (levo). Svaki slučaj korišćenja prolazi kroz sve faze. U desnoj tabeli dat je sastav projektnog tima.

Faza	Trajanje (u nedeljama)	Tim projekta:
Dizajniranje jednog slučaja korišćenja *	4 čovek/dana	1 menadžer projekta
Implementacija *	6 čovek/dana	2 softverska dizajnera (mogu i da programiraju)
Priprema testova *	1 čovek/dan	2 programera
Testiranje *	1 čovek/dan	1 tester (koji priprema i pokreće testove)
Integracija *	1 čovek/dan	1 integrator

* svaka faza se odnosi na jedan slučaj korišćenja

Napraviti projektni plan i predstaviti sve aktivnosti u Gantt dijagramu, za dva različita modela procesa:

- Projekat koristi model vodopada
- Projekat koristi inkrementalni model



Za svaki model odrediti ukupno kalendarsko vreme trajanja projekta.

PERT tehnika

- Vrlo sličan CPM tehnici
- PERT kod trajanja aktivnosti razlikuje:
 - najverovatnije vreme - vreme koje očekujemo pod normalnim okolnostima
 - optimističko vreme - najkraće vreme koje očekujemo da se završi aktivnost
 - pesimističko vreme - najgore moguće vreme, koje obuhvata sve moguće slučajnosti (izuzev nekih ekstremnih)

$$vreme = \frac{O + 4 \cdot NV + P}{6}$$



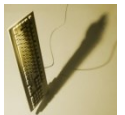
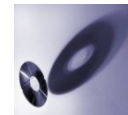
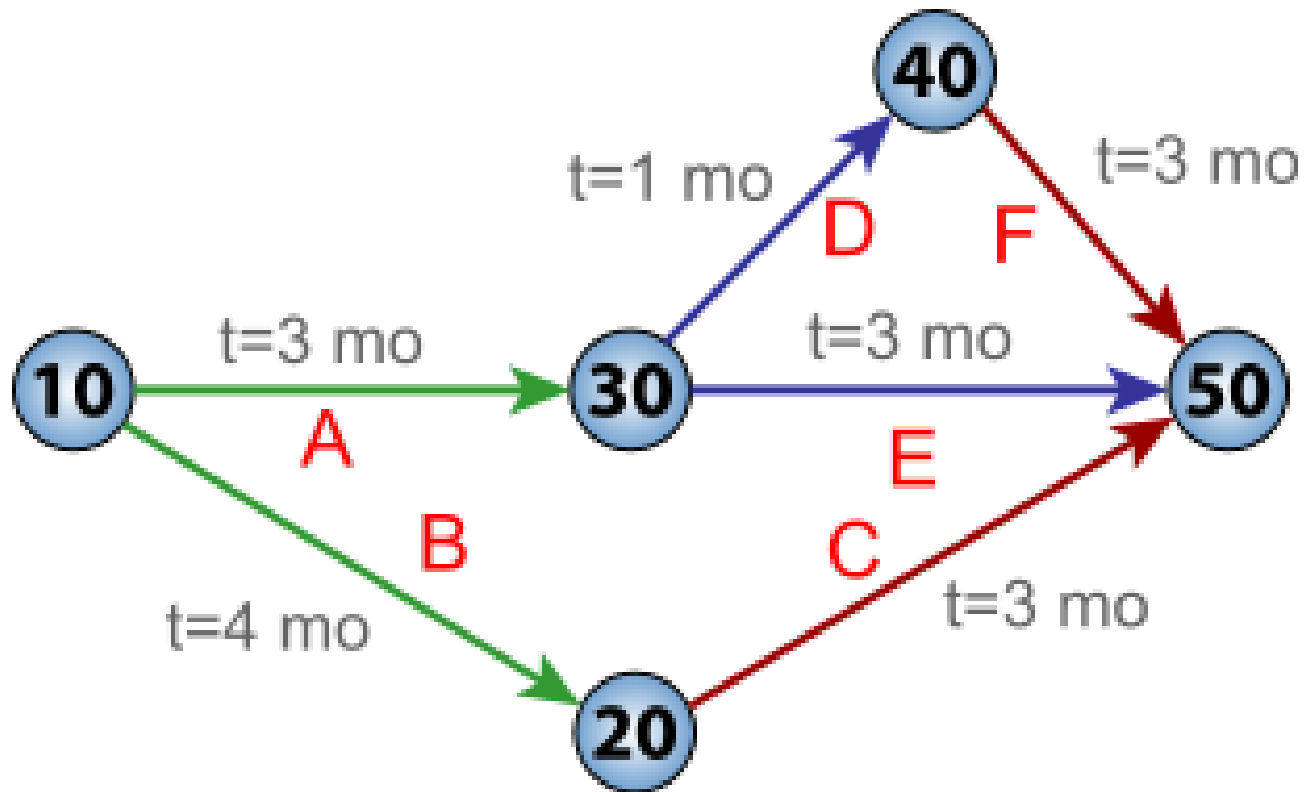
PERT tehnika

- Labela za PERT:

Redni broj	Ciljni dan
Očekivani dan	Standardna devijacija



PERT graf



Zadatak 4: PERT graf

Za dati projekat sa 7 projektnih aktivnosti (označeni u donjoj tabeli od A do G), nacrtati PERT dijagram. Očekivano vreme računati prema standardnoj formuli.

Aktivnost	Preduslov	Optimi tično	Najvero vatnije	Pesimi stično
A	/	2	4	6
B	/	3	5	9
C	A	4	5	7
D	A	4	6	10
E	B, C	4	5	7
F	D	3	4	8
G	E	3	5	8



Zadatak 4: PERT (rešenje)

Aktivnost	Preduslov	Optimistično	Najverovatnije	Pesimistično	Očekivano
A	/	2	4	6	4.00
B	/	3	5	9	5.33
C	A	4	5	7	5.17
D	A	4	6	10	6.33
E	B, C	4	5	7	5.17
F	D	3	4	8	4.50
G	E	3	5	8	5.17



Extreme programming (XP)

- Ekstremno programiranje je metodologija za razvoj softvera koja je okrenuta poboljšanju kvaliteta softvera i reakciji na promjenjene zahteve kupaca.



Neki praktični saveti - agilne metodologije

Saveti:

- Vreme između prikazivanja funkcionalnosti korisniku
- Jednostavan dizajn
- Testiranje završiti istovremeno ili malo nakon završetka kodiranja (jedinično i funkcionalno testiranje!)
- Izvršiti integraciju
- Refaktorizacija
- Programiranje u paru
- Kolektivno vlasništvo
- 40-časova nedeljno



Zaključak

- Dobro upravljanje projektom je neophodno za uspeh projekta
- Neopipljiva priroda softvera dovodi do problema u upravljanju
- Menadžeri imaju različite uloge, a najvažnije su planiranje, procena i raspored aktivnosti
- Planiranje i procena su iterativni procesi, koji se ponavljaju tokom izrade projekta



Zaključak

- Prekretnica (milestone) projekta kod CPM je predvidljivo stanje u kome se nekakav formalni izveštaj o napretku prezentuje upravi
- Rizici mogu biti projektni, proizvodni i poslovni
- Upravljanje rizicima se svodi na identifikovanje rizika koji mogu da utiču na projekat i obezbeđivanje da se ovakvi rizici ne razviju u velike pretnje

