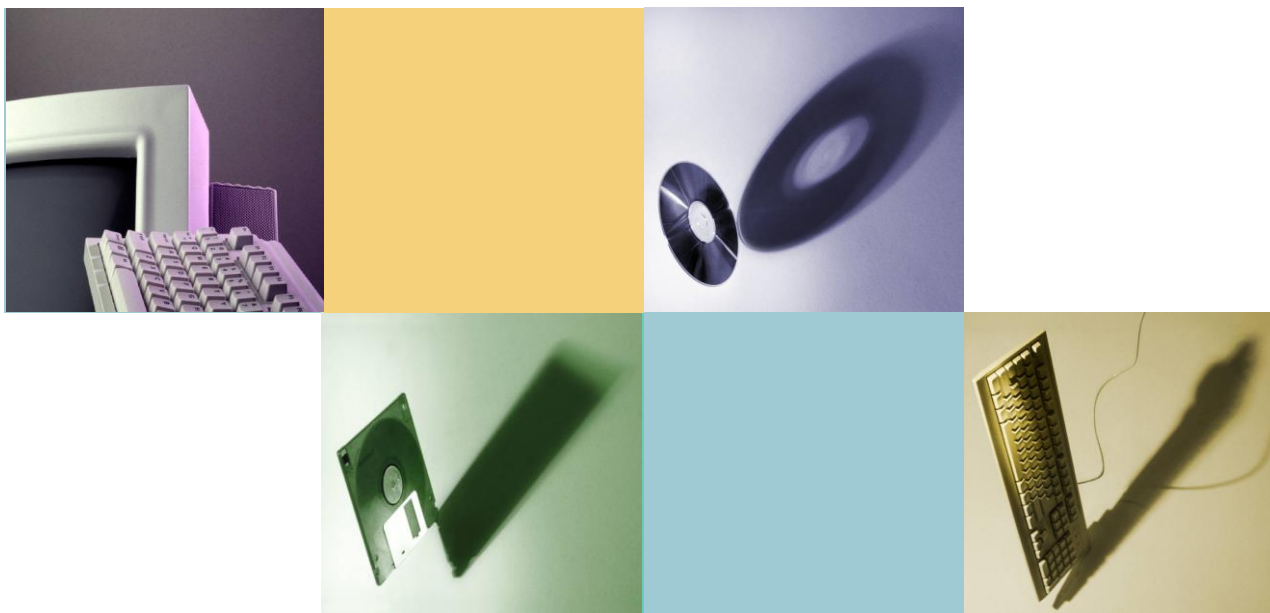


Upravljanje softverskim projektima



školska 2018/2019.

Dr Dražen Drašković, docent

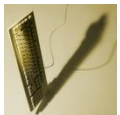
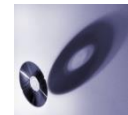
Agenda

1. Informacije o predmetu i sadržaj

2. Uvod u upravljanje projektima

3. Evaluacija projekta

4. Definisanje i planiranje projekta



O predmetu

- Naziv predmeta:
Upravljanje softverskim projektima
- izborni predmet
- Odsek: SI / RTI
- Godina: III, Semestar: 6
- Fond časova: 2 + 2 + 1
- Broj ESPB bodova: 6
- Preduslov: nema

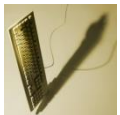


Nastavnici

- Dr Dražen Drašković, dipl. inž. el. i rač.
kancelarija 37a
drazen.draskovic@etf.bg.ac.rs (ili etf.rs)
- Predavanja i vežbe:
RTI: utorak, 12⁰⁰-14⁰⁰ (@Lola-amf)
SI: sreda, 12⁰⁰-14⁰⁰ (@308)
- Vežbe:
RTI: četvrtak, 16⁰⁰-18⁰⁰ (@311)
SI: četvrtak, 12⁰⁰-14⁰⁰ (@308)

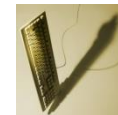


Konsultacije posle nastave i po dogovoru.



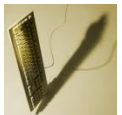
Informisanje studenata

- Predmetni sajt:
rti.etf.rs > Upravljanje softverskim projektima
- E-mail liste biće formirane na početku druge nastavne nedelje.



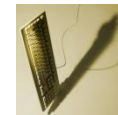
Pravila polaganja ispita

- 40% poena kroz domaće zadatke i manje projekte koji se rade u toku semestra u nekoliko faza
- 60% poena = kolokvijum + ispit
- Na pismenom ispitu (i kolokvijumu) u obzir dolazi gradivo sa predavanja, vežbi i pitanja u vezi sa domaćim zadacima



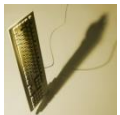
Domaći zadaci

- Zadaci su obavezni i ukupno nose 60 poena.
- Domaći zadaci mogu da se rade samo za junsko-julski i septembarski ispitni rok, u grupama od po 2 člana i usmeno se brane u određenim terminima nakon kolokvijuma i ispita.
- Jednom odbranjeni zadaci važe u svim ispitnim rokovima tekuće školske godine.
- Napomena: samo u JUNU i SEPTEMBRU!!!



Sadržaj predmeta (1)

- 1) Uvod u upravljanje softverskim projektima
- 2) Planiranje projekta
 - Projektni plan
 - Studija slučaja (Business case)
 - Analiza zahteva
 - Analiza troškova i dobiti
- Step Wise project management method
- 3) Upravljanje timom
 - načini rada u timu
 - organizacija tima i donošenje odluka
 - uloge i odgovornosti u softverskom timu
 - identifikacija i dodeljivanje uloga
 - praćenje projekta kroz aktivnosti
 - “team problem resolution”

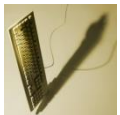


Sadržaj predmeta (2)

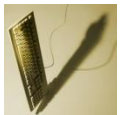
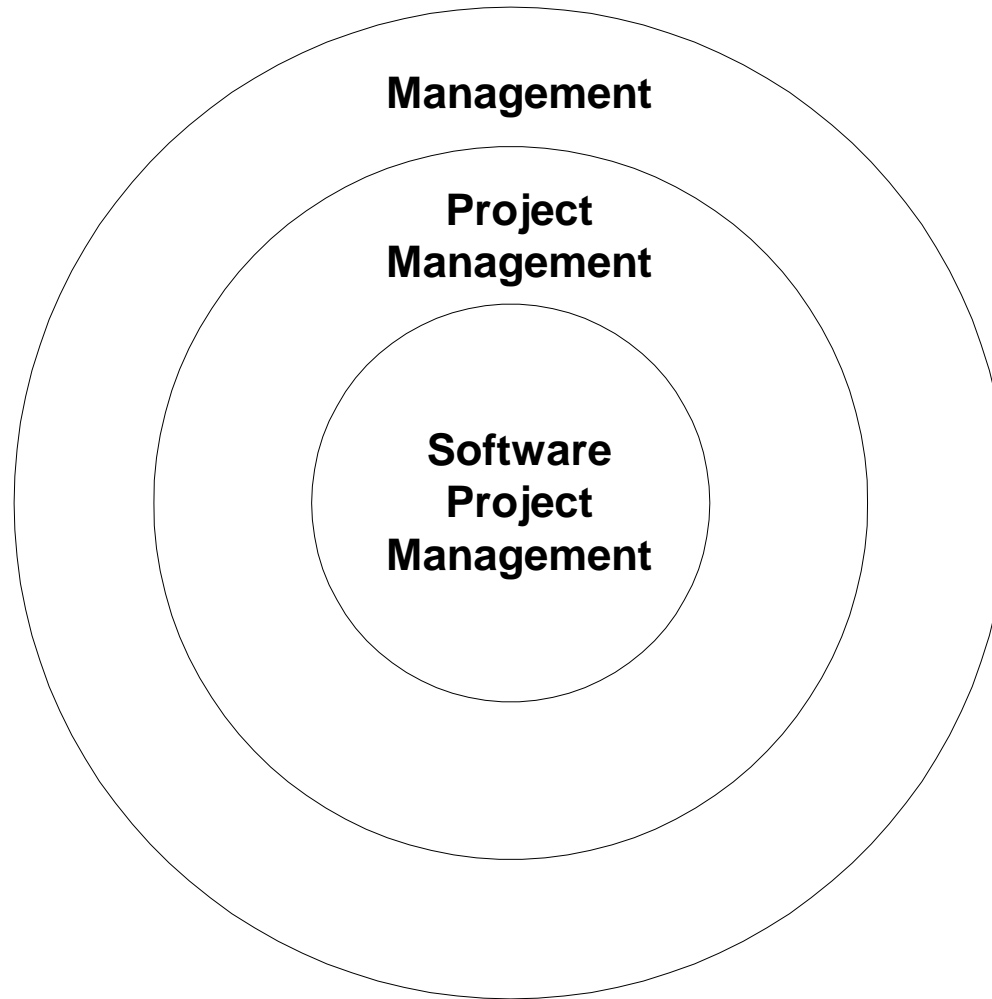
- 4) Upravljanje ljudskim resursima i komunikacijom na projektu
- 5) Upravljanje rizicima i analiza rizika
- 6) Sigurnost kvaliteta
- 7) Kontrola i monitoring projekta
- 8) Upravljanje softverom i kontrola verzija

Čućete još:

- Alati za upravljanje projektom (Microsoft Project, JIRA)
- Alati za upravljanje kupcima (SugarCRM)
- CMMI model (Capability Maturity Model Integration)
- Modeli razvoja softvera i proces razvoja softvera



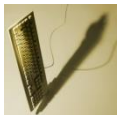
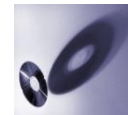
Software project management



Zašto je značajan

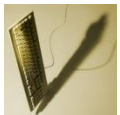
Software project management?

- Oblast koju moraju da nauče studenti SE i CS
- A zbog čega je značajna oblast?
- Zbog novca!!!
- UK, 2002-2003:
 - 2.3 milijardi funti za ICT projekte
 - 1.4 milijarde funti za projekte infrastrukture
 - Ministarstvo za rad i penzije uložilo 800 miliona funti u ICT
- Države članice EU (FP7): € 9.1 milijardi izdvojile za ICT projekte
- Planirano za ciklus H2020 uložiti € 80 milijardi u projekte (oko 20-25% u ICT)



Šta je projekat? (1)

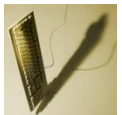
- Def. Projekat = planiranje aktivnosti
- PMI Def. = privremeno ulaganje napora da bi se stvorio jedinstveni proizvod ili usluga
- Šta je planiranje?
- Planiranje kod istraživačkih projekata može biti vrlo teško
- Planiranje - pažljivo misliti o nečemu, pre nego što to uradimo
- Kod neizvesnih projekata: sve dok su ispunjeni planovi i vidljivi rezultati



Neki projekti...

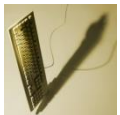


ROVIO = 51 poluuspešan ili neuspešan projekat



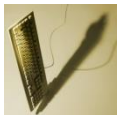
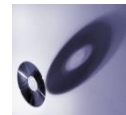
Šta je projekat? (2)

- Planiranje aktivnosti mora biti dokumentovano, zbog konzistentnosti i novih članova tima
- Da li postoji razlika kada planirate:
 - novi projekat
 - projekat koje je sličan prethodnom sistemu koji ste razvijali



Upravljanje softverskim projektima

- Organizacija, planiranje i raspoređivanje aktivnosti na softverskim projektima.
- Bavi se aktivnostima koje treba da obezbede da se softver isporuči na vreme i po planu i da bude u skladu sa zahtevima korisnika (ili onoga ko finansira izradu).
- Upravljanje softverskim projektima je neophodno jer je razvoj softvera uvek podložan ograničenjima (vremenskim i finansijskim) koje postavlja onaj ko finansira razvoj softvera.



Aktivnosti u upravljanju projektima

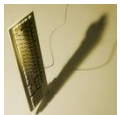
Rutinski posao

Neizvesnost
ishoda

Posao

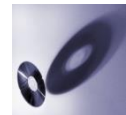
Projekti

Istraživanje



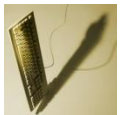
Karakteristike za upravljanje projektima

- Liderstvo
- Komunikacione veštine
- Sposobnost rešavanja problema
- Sposobnost pregovaranja
- Uticaj na organizaciju
- Mentorski rad
- Proces upravljanja i tehnička ekspertiza



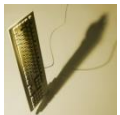
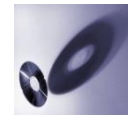
Karakteristike projekata (1)

- Sledeće karakteristike odlikuju projekte:
 - planiranje je potrebno i neophodno
 - uključiti i ne-rutinske zadatke
 - ispuniti određene ciljeve ili napraviti određen proizvod na kraju
 - projekat unapred vremenski određen
 - radi se za nekoga, a ne samo za sebe
 - posao obuhvata nekoliko specijalizovanosti
 - ljudi se grupišu u privremene radne grupe koje izvršavaju neke zadatke



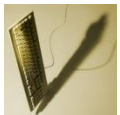
Karakteristike projekata (2)

- posao se uvek deli u nekoliko faza
- resursi, koji su dostupni za korišćenje na realizaciji projekta, su ograničeni
- projekat je najčešće veliki i/ili kompleksan
- Vrlo je važna i veličina projekta
- Projekat koji zapošljava 20 programera ili projekat koji zapošljava 10 programera?



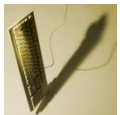
Vežba #1 - Projekti

- proizvodnja novina
- robot na Marsu koji traži tragove života
- organizacija svadbe
- jedinstveni finansijski sistem koji podržava evropsku valutu
- istraživački projekat Smarter Planet 2020.
- pisanje operativnog sistema za novi računar
- instaliranje nove verzije Word-a 2013 u studentskoj službi



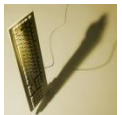
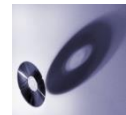
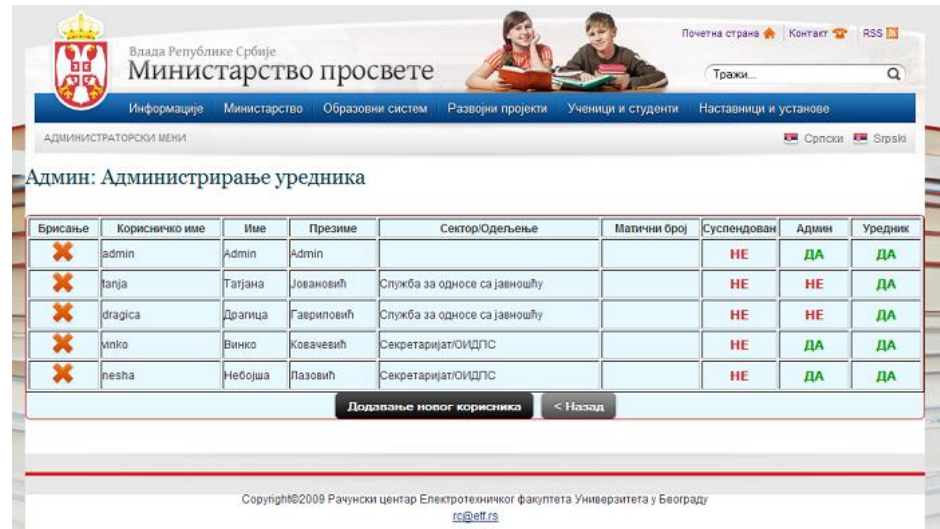
Opšte stvari u upravljanju

- Pisanje predloga, studije izvodljivosti, planiranje i raspoređivanje aktivnosti, procena troškova, praćenje statusa i inspekcije, izbor i procena kadrova, pisanje i prezentovanje izveštaja,...
- Nabrojane aktivnosti nisu specifične za upravljanje softverskim projektima.
- Mnoge tehnike upravljanja drugim tehničkim projektima su podjednako primenljive na upravljanje softverskim projektima.
- Kompleksni tehnološki sistemi imaju slične probleme kao i softverski sistemi.



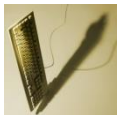
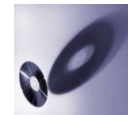
Softverski projekti vs drugi projekti

- Nevidljivost
- Složenost (kompleksnost)
- Prilagođenost
- Fleksibilnost



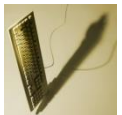
Učesnici softverskog projekta

- Kod svakog softverskog projekta postoje sledeće kategorije učesnika (*stakeholders*):
 - Senior menadžeri (*Senior Project Managers*), koji definišu poslovna pitanja koja često imaju uticaj na projekat
 - Menadžeri projekta (*Project Managers*), koji moraju da planiraju, motivišu, organizuju i kontrolišu stručnjake koji prave softver
 - Rukovodioci i timovi stručnjaka (*Executives & Teams*), koji vladaju tehnikom neophodnom za razvoj softverskog sistema (izvođači radova)
 - Kupci (*Customers*), koji su odredili zahteve po kojima se softver projektuje
 - Krajnji korisnici (*Clients*), koji koriste softver tek nakon što je predat na korišćenje
 - opciono: Sponzor projekta (*Project sponsor*)



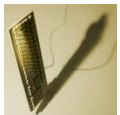
Moguće uloge projektnog menadžera

- Project Administrator / Coordinator
- Assistant Project Manager
- Project Manager / Program Manager
- Executive Program Manager
- Vice President of Program Development
- ...

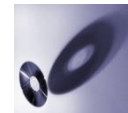
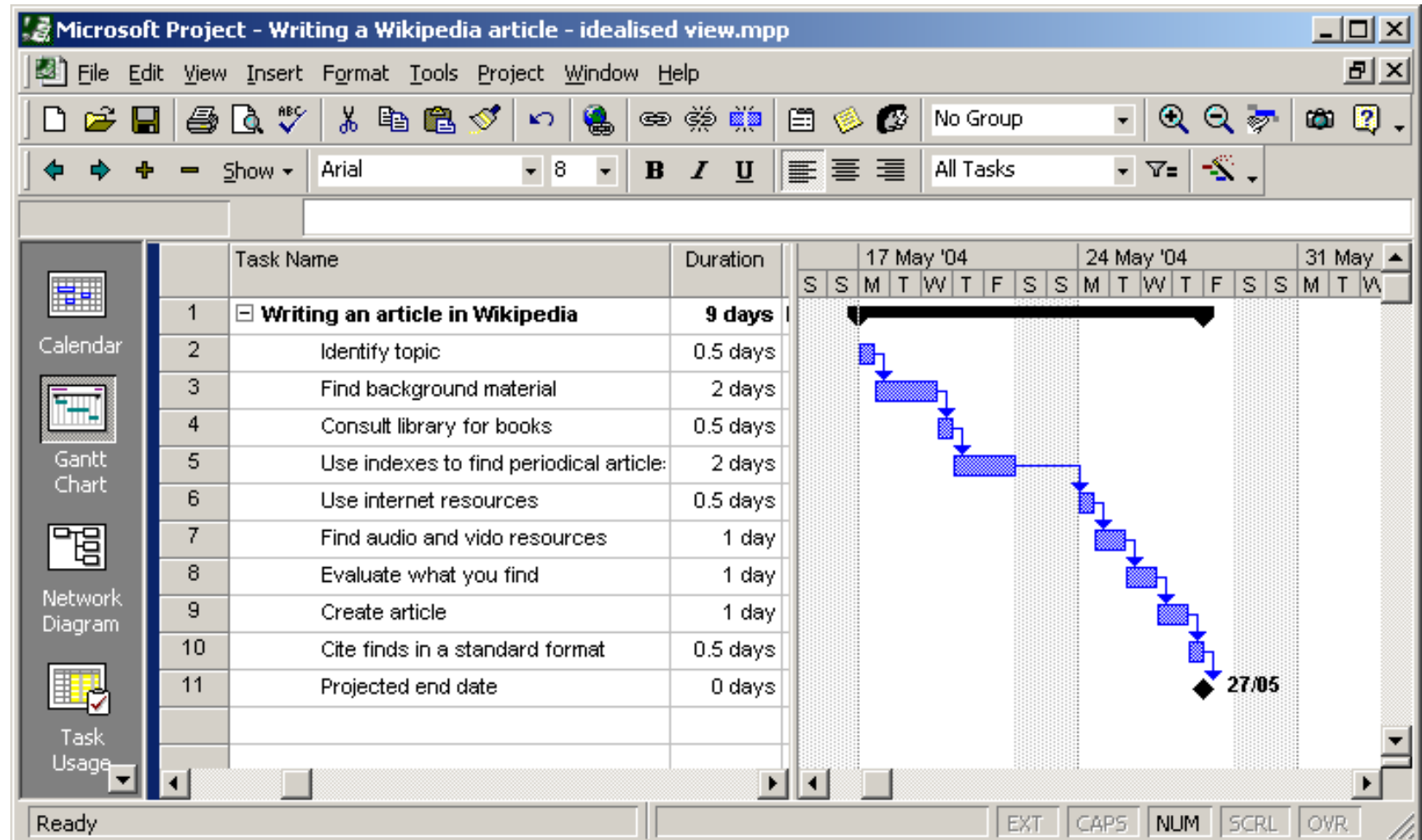


Alati za upravljanje projektima (1)

- Projekti niske klase
 - Osnovne funkcionalnosti, upravljanje zadacima, rad sa grafikonima
 - Primeri:
MS Excel, Milestones Simplicity (tzv. project scheduler-i)
- Projekti srednje klase
 - Rad na većim projektima i sa više projekata istovremeno, potrebna analiza podataka u projektu/projektima
 - Primer: MS Project (oko 50% tržišta koristi)
- Projekti visoke klase
 - Veoma veliki projekti, sa specijalizovanim potrebama
 - Primeri:
Primavera Project Manager, AMS Realtime



Alati za upravljanje projektima (2)



Microsoft Excel

Predlog troskova v2.xls [Compatibility Mode] - Excel

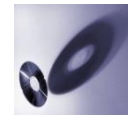
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW ADD-INS ABBYY FineReader 11 PDF Architect 4 Creator TEAM YuC

Clipboard Font Alignment Number Styles

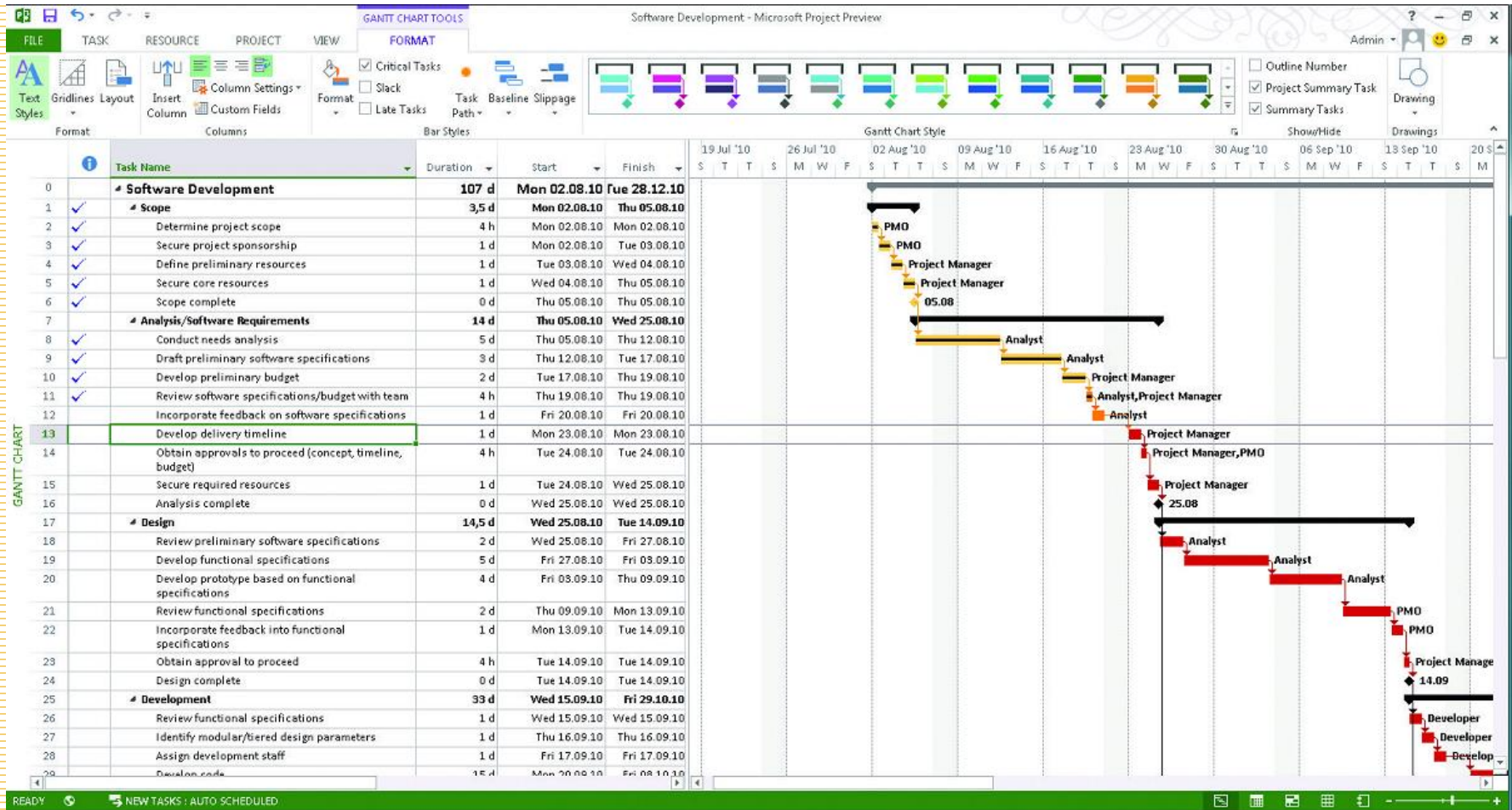
Calibri 11 A A B I U Wrap Text Merge & Center General \$ % , .00 .00 Conditional Formatting Table Normal Neutral

N28 X ✓ fx

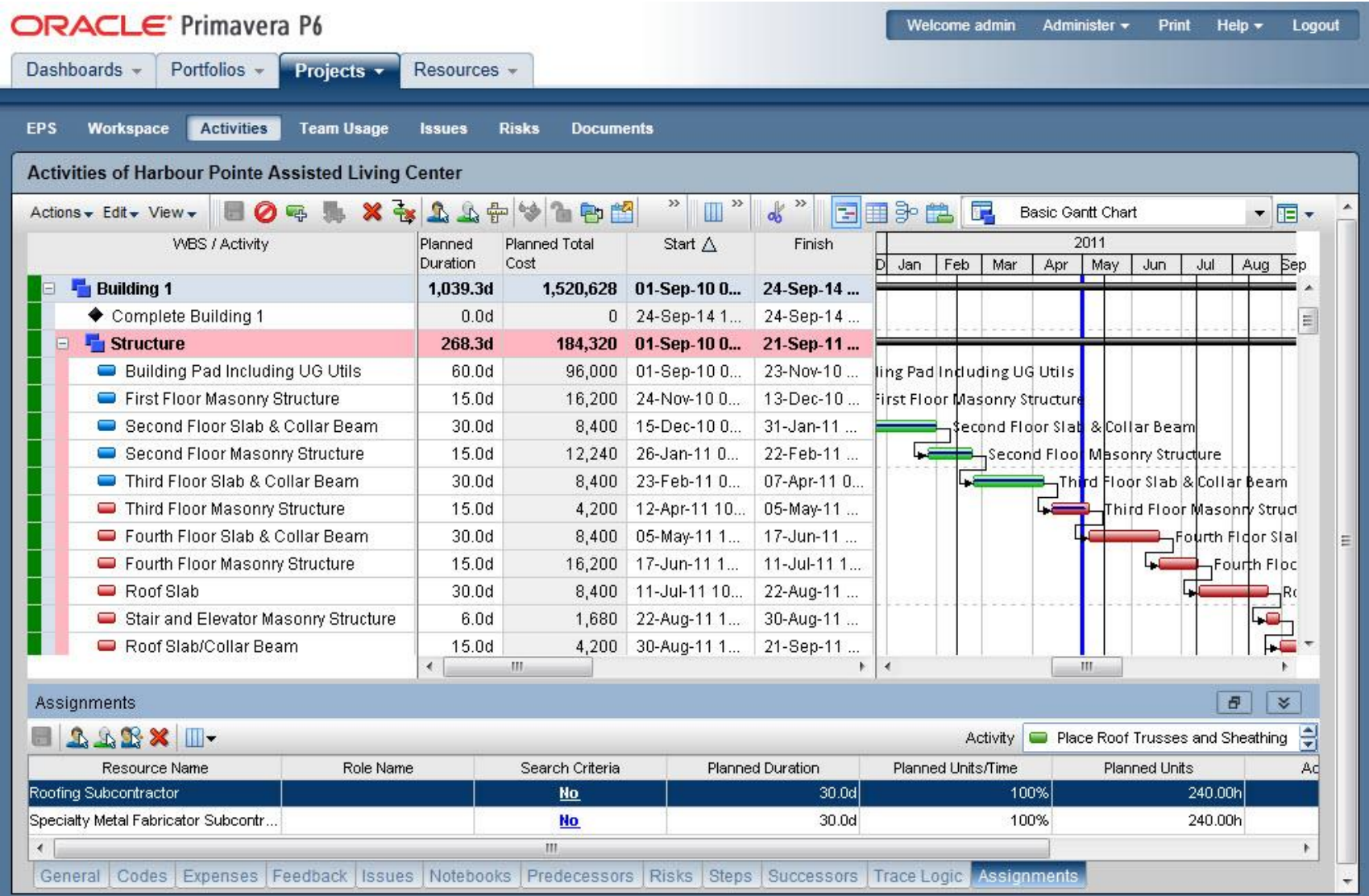
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
9															
10			ETF	Company1	Company2	Company3	Company4	SUM po WP	Procenat						
11		WP1	4	30	2	2	2	40	11%	132520					
12		WP2	24			11	9	44	12%	110720					
13		WP3	42	22	3			67	19%	196760					
14		WP4	20	26				46	13%	142600					
15		WP5	24		15			39	11%	118920					
16		WP6		40	4			44	12%	155200					
17		WP7	12	6	2	16	13	49	14%	130360					
18		WP8	5	8	1	4	12	30	8%	79900					
19		Ukupno po partneru	131	132	27	33	36	359	359	1066980					
20		Preostalo po partneru	0	0	0	-1	0								
21															
22															
23															
24															
25															



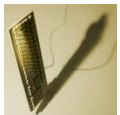
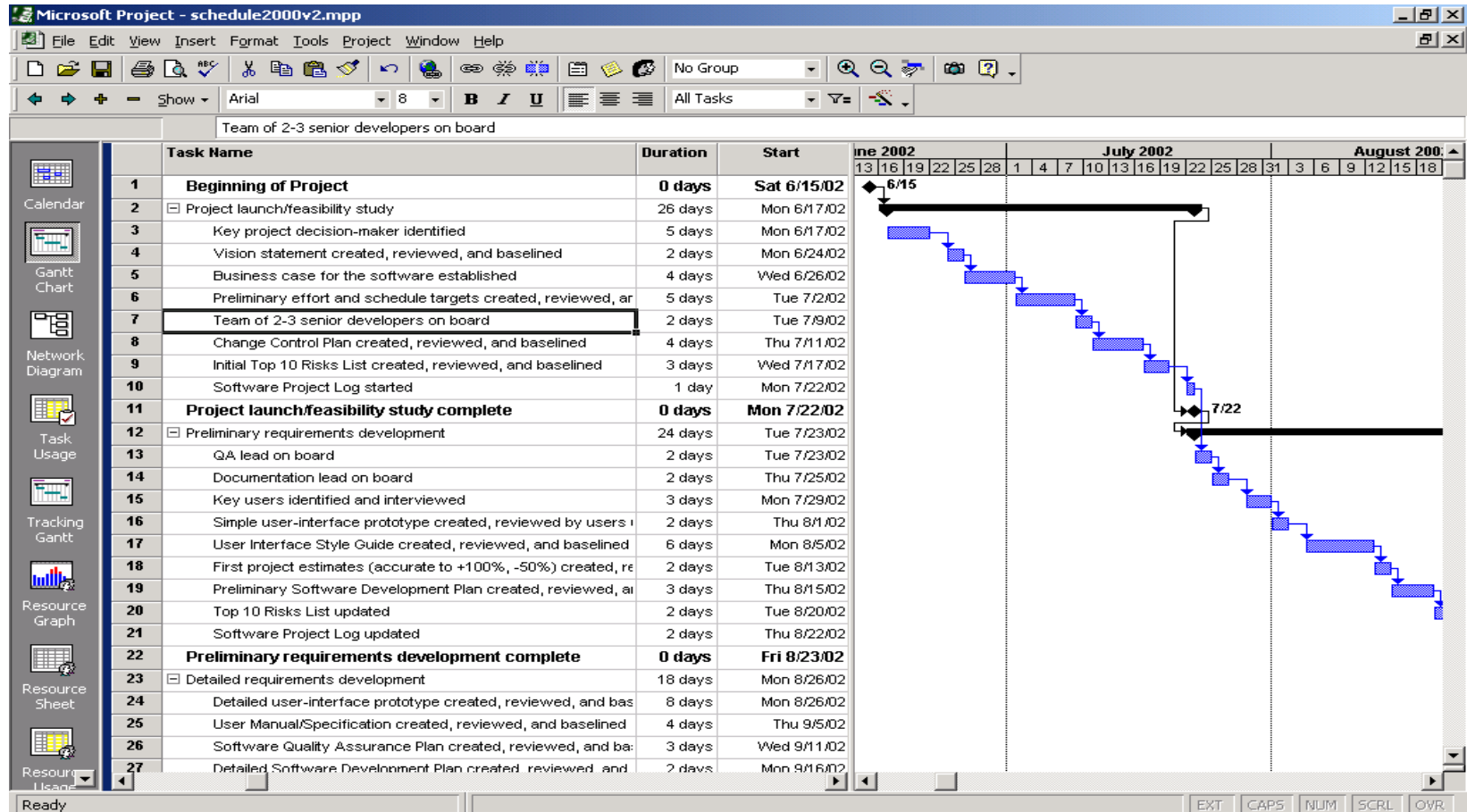
Microsoft Project



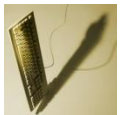
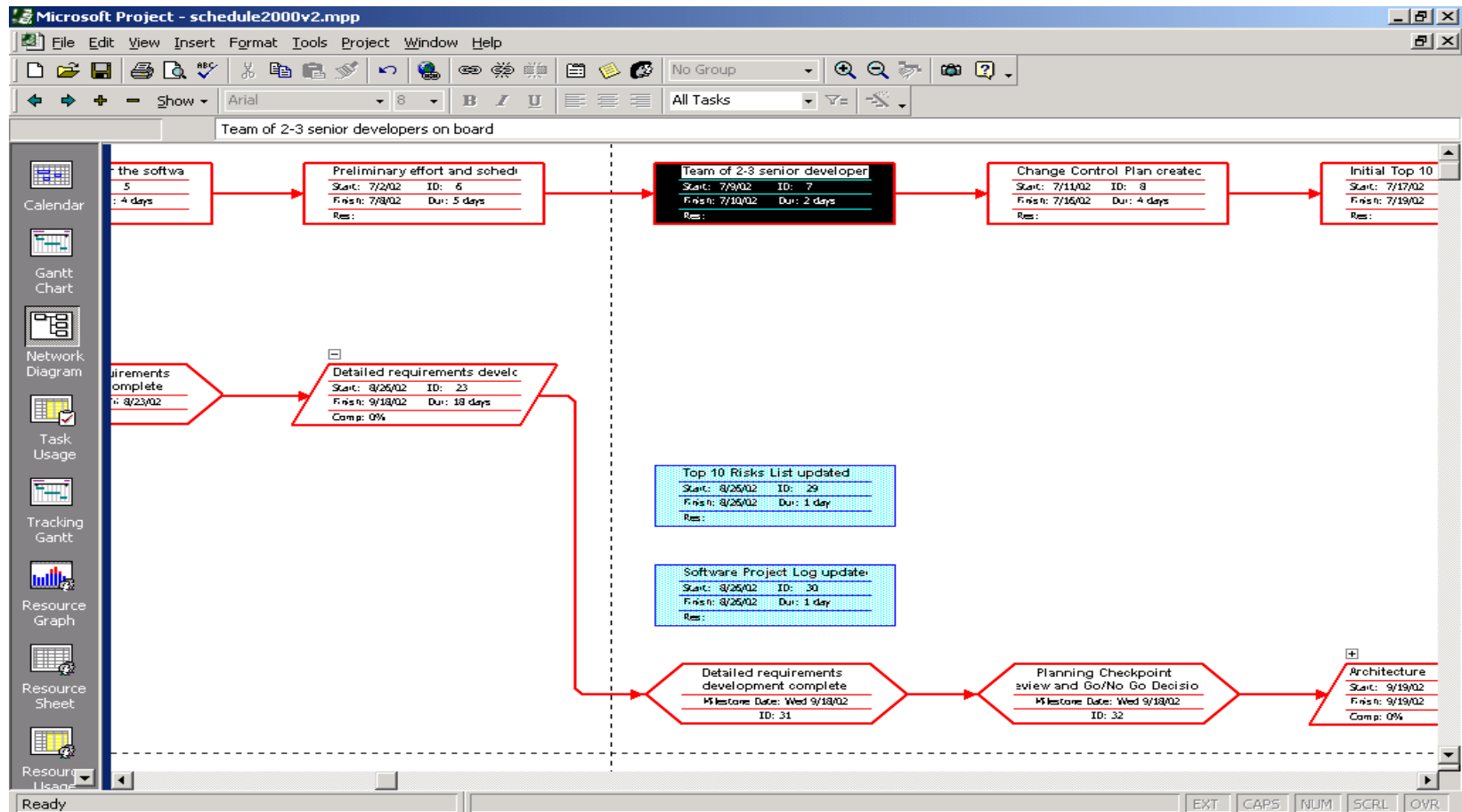
Primavera P6 Enterprise Project Portfolio Management



Gantt diagram

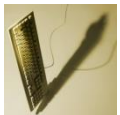
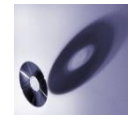
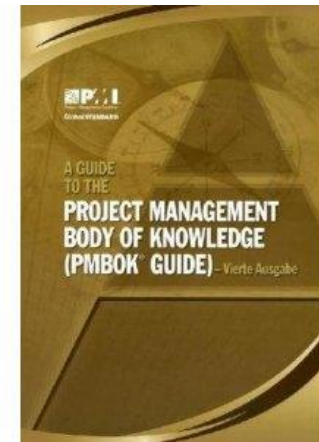


Dijagram mreže aktivnosti



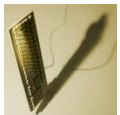
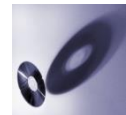
Project Management Institute (1)

- PMI (osn. 1969) - svetska organizacija za upravljanje projektima
- PMBOK (Body of Knowledge) grupiše 47 procesa u 5 osnovnih procesnih grupa:
 - iniciranje projekta,
 - planiranje projekta,
 - izvršavanje projekta,
 - monitoring i kontrola projekta,
 - završetak projekta.



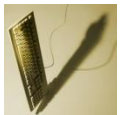
Project Management Institute (2)

- Deset oblasti znanja:
 - upravljanje integracijom projekta (Project Integration Mngm.)
 - upravljanje obimom (Project Scope Mngm.)
 - upravljanje vremenom (Project Time Mngm.)
 - upravljanje troškovima (Project Cost Mngm.)
 - upravljanje kvalitetom (Project Quality Mngm.)
 - upravljanje ljudskim resursima (Project Human Resource M.)
 - upravljanje komunikacijom (Project Communications M.)
 - upravljanje rizicima (Project Risk Management)
 - upravljanje nabavkama (Project Procurement Management)
 - upravljanje učesnicima projekta (Project Stakeholders M.)

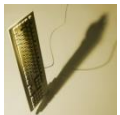
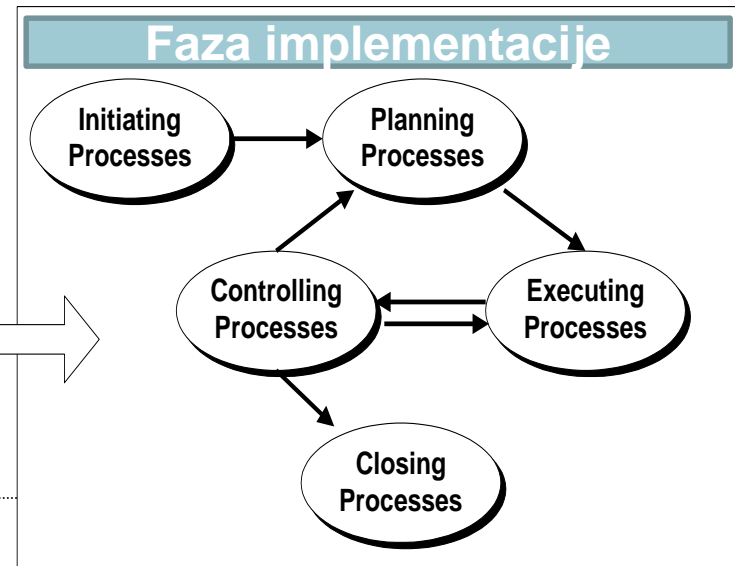
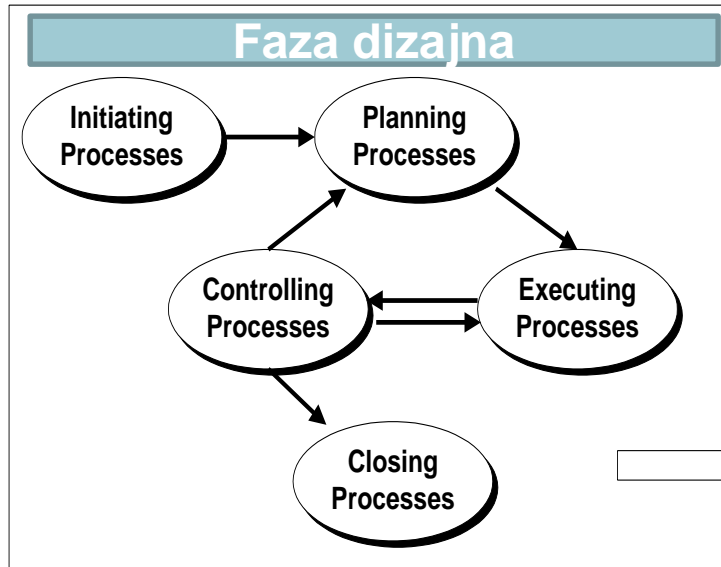
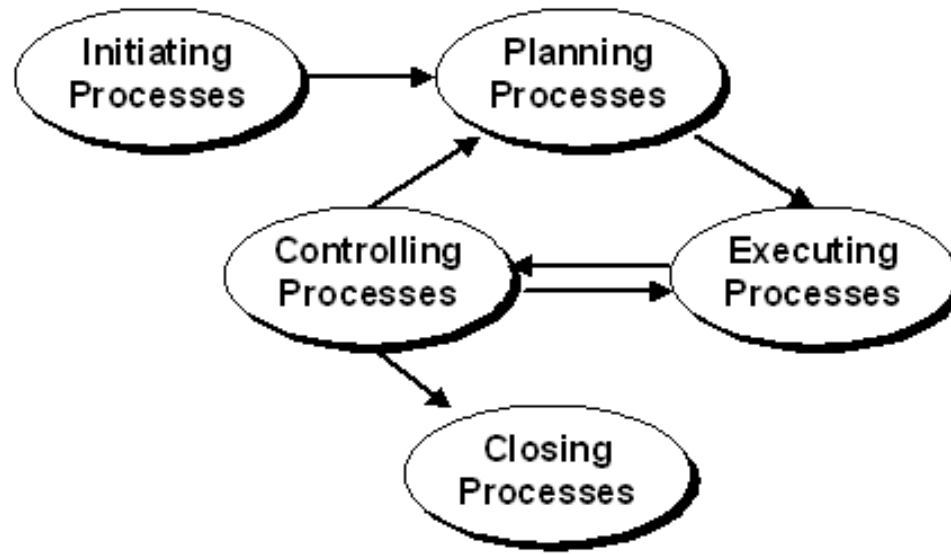


5 PMI procesa

- 1. Iniciranje (pokretanje) projekta
- 2. Planiranje projekta
- 3. Izvršavanje projekta
- 4. Kontrolisanje projekta
- 5. Zatvaranje projekta
- Napomena: mogu da se ponavljaju za svaku fazu
- Svaki proces je opisan kroz:
 - Ulaze
 - Alate i tehnike
 - Izlaze

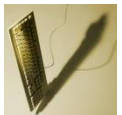


Veze između procesa (1)



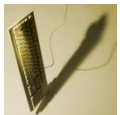
PMI - Iniciranje

- Ulazi:
 - Opis soft. proizvoda
 - Strateški plan
 - Kriterijumi za izbor projekta
 - Istorijat
- Izlazi:
 - Pravo na projekat
 - Dodeljen menadžer projekta
 - Ograničenja
 - Pretpostavke



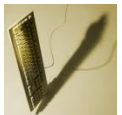
PMI - Planiranje

- Obuhvata:
 - Planiranje obima
 - Definirati obim
 - Dijagram aktivnosti
 - Definirati podaktivnosti
 - Procena trajanja aktivnosti
 - Planiranje resursa
 - Procena budžeta
 - Formiranje budžeta
 - Planiranje rizika
 - Raspored razvoja projekta
 - Planiranje kvaliteta
 - Planiranje komunikacije
 - Planiranje organizacije
 - Nalaženje ljudi za projekat
 - Planiranje nabavki
 - Razvoj projektnog plana



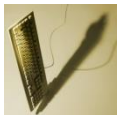
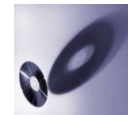
PMI - Izvršavanje

- Obuhvata:
 - Izvršavavanje projektnog plana
 - Verifikacija obima posla
 - Osiguranje kvaliteta
 - Razvoj tima
 - Distribucija informacija
 - Administracija ugovora



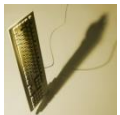
PMI - Kontrolisanje

- Obuhvata:
 - Celokupnu kontrolu
 - Kontrolu promene obima
 - Kontrolu rasporeda
 - Kontrolu troškova
 - Kontrolu kvaliteta
 - Kontrolu rizika
 - Izveštavanje o projektu
- Završetak projekta obuhvata:
 - administrativno zatvaranje i zaključivanje ugovora



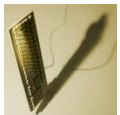
Četiri dimenzije projekta (1)

- 1) Ljudi
 - Izbor tima
 - Organizacija tima
 - Motivacija
 - Drugi faktori (dodeljivanje zadataka ljudima, razvoj karijere, balans individualnog i timskog posla...)
- 2) Proces
 - 2 tipa: upravljački i tehnički
 - Planiranje životnog ciklusa, osiguranje kvaliteta, upravljanje rizicima,...
 - Orijentacija na kupce
 - Poboljšanje procesa zrelosti (npr. CMMI)
 - Izbegavanje dorada (ponovnog rada)

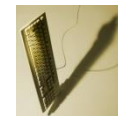


Četiri dimenzije projekta (2)

- 3) Proizvod
 - Zahtevi korisnika u saglasnosti sa proizvodom
 - Upravljanje dimenzijama (veličinom) proizvoda
- 4) Tehnologija
 - Često najmanje važna dimenzija
 - Izbor jezika i alata
 - Vrednost i troškovi ponovne upotrebe

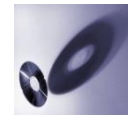


TIMОВI



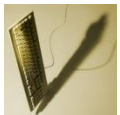
Timski rad

- Mnogi softverski sistemi su suviše veliki ili kompleksni da bi bili razvijeni od strane pojedinca
=> stvaraju se softverski timovi!
- Zadaci po svojoj prirodi mogu biti:
 - Deljeni
 - Individualni
 - Kombinovani



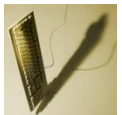
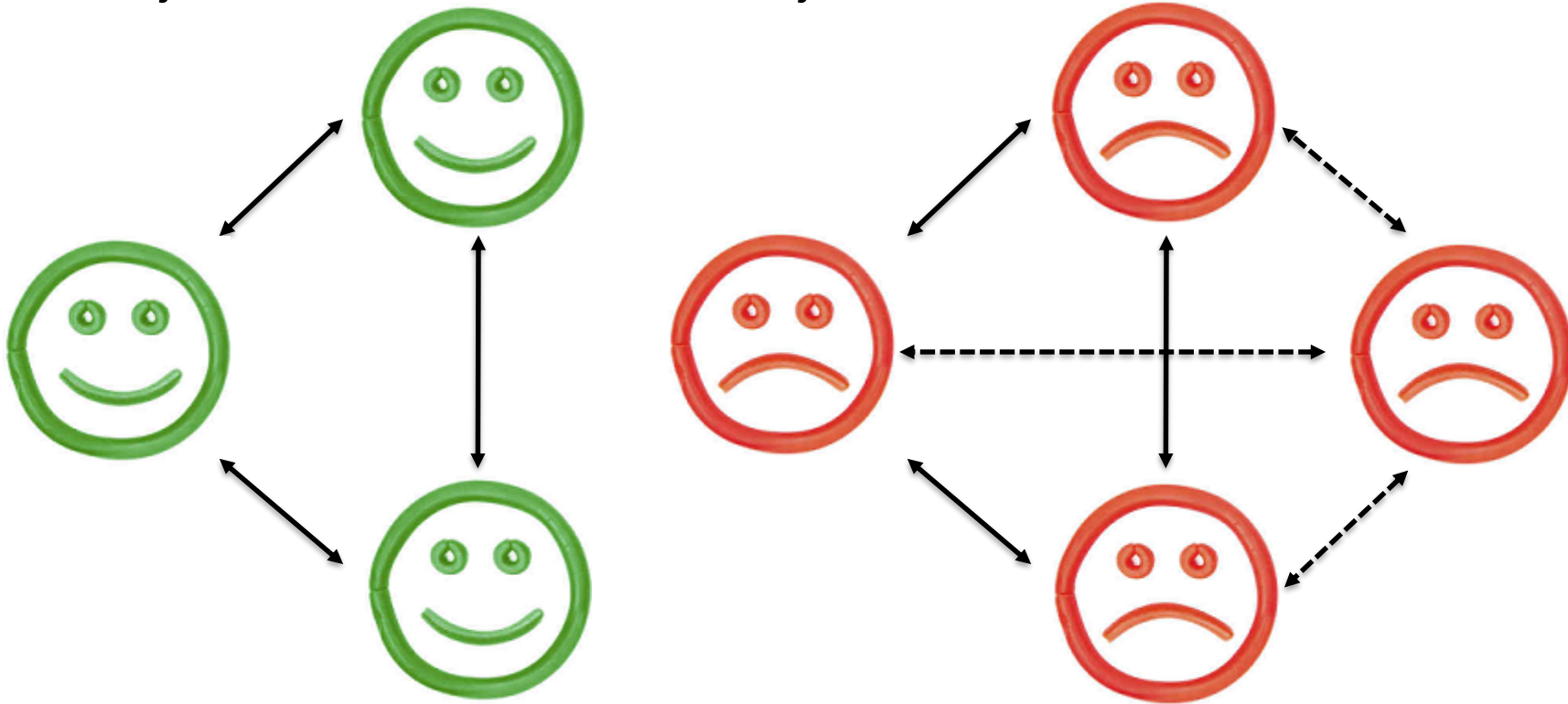
Timski rad i efikasnost

- Da bi bio efikasan, projektni tim mora biti organizovan tako da maksimizuje veštine i sposobnosti svakog člana tima.
To je posao tim lidera.
- Mana:
Grupisanje ljudi, koji rade na nekom zadatku, vrlo često utiče negativno na autoritet postojećeg sistema u organizaciji.
- Prednost:
Tim obično sadrži specijaliste za različite oblasti, koji su usmereni da završe neki važan zadatak.



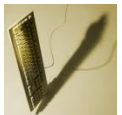
Komunikaciona eksplozija

- Ako svako u timu od n ljudi, jedna osoba komunicira sa svakom drugom osobom iz tima, broj komunikacionih kanala je $= 0.5 \times n \times (n-1)$



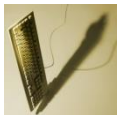
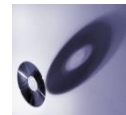
Različite strukture timova

- Demokratski tim
- Glavni-programer tim („šefovski tim“)
- Modifikovani šefovski tim
- Sinhronizovan i stabilizovan tim
- Ekstremno programiranje (XP)



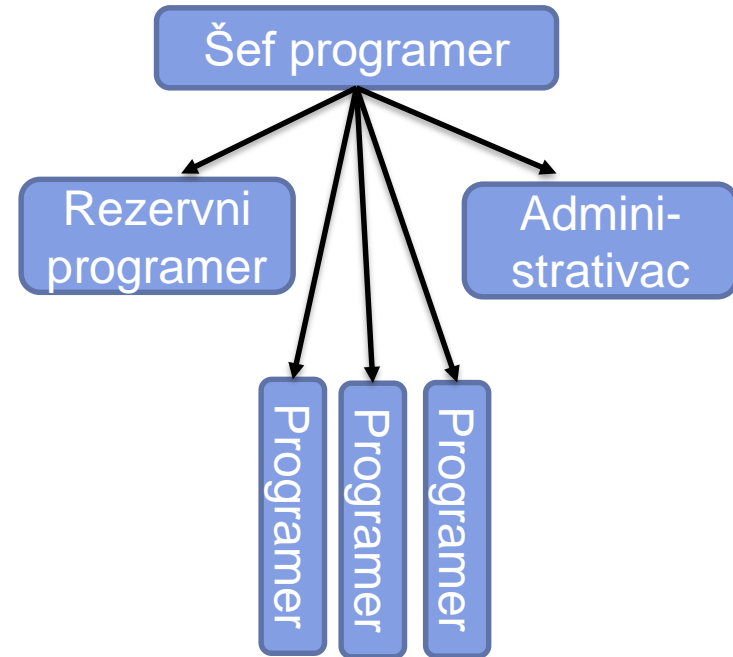
Demokratski timovi

- Zahtevaju programiranje bez ega:
 - Programeri nisu previše vezani za svoj kod
 - Neophodna dobra atmosfera u timu i međusobna saradnja
 - Pronalaženje bagova daje pozitivan efekat
- 10 programera „bez ega“ = demokratski tim
- Svako je jednak i tim je samoorganizovan
- Ne postoji imenovani lider => u suprotnosti sa konvencionalnim principom upravljanja
- Teorija tvrdi da ovakva grupa radi dobra za kompleksne probleme

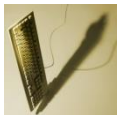


Šefovski tim

- Po uzoru na hiruške timove: hijerarhija se koristi da bi se prevazišla komunikaciona eksplozija, a specijalizacija da unapredi produktivnost
- Šef-programer tim od 6 osoba redukuje broj komun. kanala sa 15 na 5

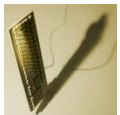


- Uloge:
 - Šef (glavni programer): dizajnira arhitekturu sistema, programira kompleksne delove koda i integriše kod u celinu, sveobuhvatni menadžer
 - Rezervni (zamenik) programer: preuzima po potrebi zadatke od šefa, dizajnira testove
 - Administrativac: održava bazu podataka i dokumentaciju, pokreće testove



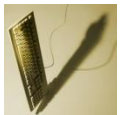
Šefovski tim – istorijat

- Rezultat inicijalnog test projekta je bio izvanredan:
 - „NewYork Times“ alat za automatsko sečenje fajla
 - 83 000 LOC za 22 meseca (11 čovek/godina)
 - Polovina modula (svaki od po 200-400 LOC) bio je korektan prilikom prvog kompajliranja
 - Alat za održavanje fajlova bio operativan 20 meseci bez greške
 - Nakon ovoga nije postojao nijedan sličan uspešan projekat!
 - Razlog: šef Terry Baker je jedan od najvećih programera za superračunare



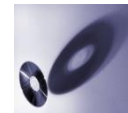
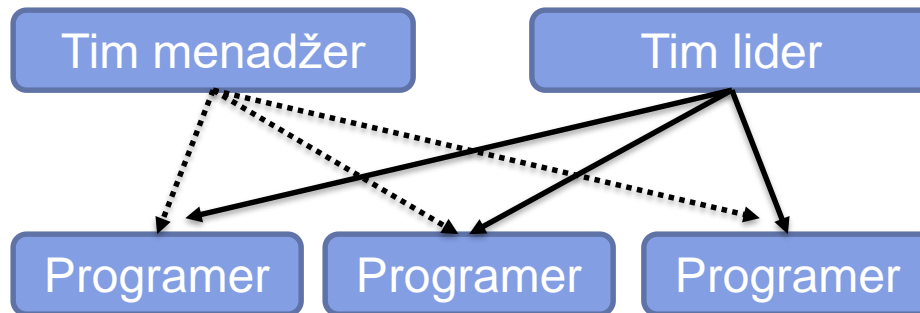
Problemi kod šefovskog tima

- Dobrog šefa je vrlo retko naći:
treba da bude presek visoko obrazovnog programera i
veoma uspešnog menadžera
- Zamenika glavnog programera je još teže naći:
kapacitet kao šef, ali spreman da radi za manju platu i
da radi u podređenom položaju (na nižoj poziciji)
- Administrativca je takođe teško naći:
Ko želi da ne radi ništa oko razvoja softverskog sistema,
nego samo da radi na dokumentaciji svakog dana.



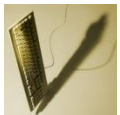
Modifikovani šefovski tim

- Šef programer podeljen u dve uloge:
 - Tim lider (za tehnička pitanja)
 - Tim menadžer (za netehnička menadžerska pitanja)
- Pažljivo odvojene odgovornosti uloga u timu, kako ne bi bilo konflikta interesa
- Ali mogu postojati neki delovi preklapanja (recimo kada jedan ode na godišnji odmor)
- Struktura tima može da se uveća ako se uvede još liderskih nivoa (viši šefovi u hijerarhiji)



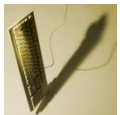
Sinhronizovan i stabilizovan tim

- Microsoft model; uspešan za veoma velike projekte
 - Više od 3000 programera i testera je radilo na Win 2000
- Mali timovi od 3-8 programera i 3-8 testera, rade u paraleli
- Pojedincima je dozvoljena sloboda da dizajnira i implementiraju specifikaciju, ali:
 - Programski kod mora biti integrisan na dnevnom nivou
 - Ukoliko tvoj programski kod sprečava kompilaciju, on mora da bude fiksiran odmah
- Microsoft ima svoju poslovnu etiku već godinama, tako da njihov tim možemo nazvati „tim jake korporativne kulture“



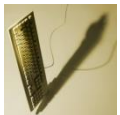
XP tim

- Baziran na programiranju u paru:
 - šire znanje koje imaju dve osobe nego pojedinac
 - pomaže manje iskusnim programerima da se ubrzaju
- Dokazi:
 - Istraživanje na Univerzitetu Utah na naprednom kursu programiranja:
Svoje testove parovi su radili za 60% vremena od vremena koje je bilo potrebno pojedincu.
Test je prošlo 94% studenata, za razliku od 78% kada su radili pojedinačno.
- Programiranje u paru poboljšava zadovoljstvo poslom i sveukupno poverenje.



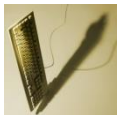
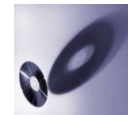
Vežba 2: Odaberite strukturu tima

- Kao projektni menadžer treba da odaberete strukturu tima za sledeće projekte:
 - 1) Projekat za kvantne računare ima 5 istraživača. Ne postoji strogi rok za završetak. Ako uspe na ovom projektu, ovaj tim će nastaviti da radi i na kasnijim projektima.
 - 2) Projekat razvoja sistema za naplatu za rudarsku korporaciju. Ovaj projekat angažuje 30 čovek/godina i mora biti završen u roku od 10 meseci (striktno)
- Rešenja?
 - 1) demokratski ili XP tim
 - 2) modifikovani šefovski tim



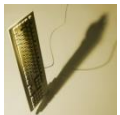
Strukture tima - kategorije

- Struktura tima zavisi od stila upravljanja vašom organizacijom, broja ljudi koji će činiti tim, njihovih nivoa znanja, kao i ukupne težine problema na kome se radi.
- Postoje 3 generičke strukture tima:
 - Democratic decentralized (DD)
 - Controlled decentralized (CD)
 - Controlled centralized (CC)



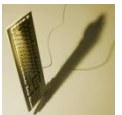
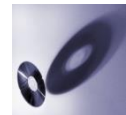
DD struktura tima

- Demokratski decentralizovani timovi
- Ovaj softverski tim nema stalnog lidera.
- Tim vode koordinatori tokom trajanja nekog zadatka, a zatim budu zamenjeni drugim, koji mogu voditi različite zadatke.
- Odluke se donose grupnim koncenzusom.
- Komunikacija i kontrola su horizontalni.



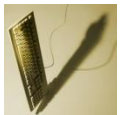
CD struktura tima

- Kontrolisani decentralizovani timovi
- Ovaj softverski tim ima tim lidera koji koordinira određenim zadacima i druge vođe koji imaju odgovornost za podzadatke.
- Rešavanje problema ostaje timska aktivnost, ali implementacija rešenja se deli među podgrupama od strane tim lidera.
- Komunikacija među podgrupama je horizontalna, ali postoji i vertikalna komunikacija u kontrolnoj hijerarhiji.

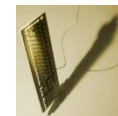
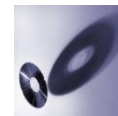


CC tim

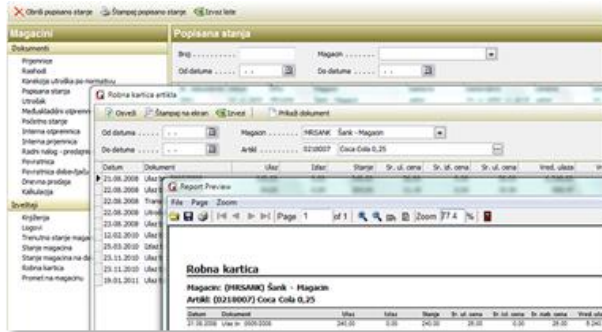
- Kontrolisani centralizovani timovi
- Tim lider se bavi unutrašnjom koordinacijom u timu.
- Rešavanje problema takođe na najvišem nivou.
- Komunikacija između lidera i članova tima je vertikalna.



EVALUACIJA PROJEKTA



Trougao kompromisa



PROIZVOD

Cilj, performanse,...



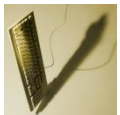
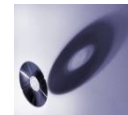
RASPORED

Vreme



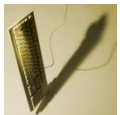
TROŠKOVI

Resursi



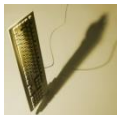
Preduzetništvo

- Delatnost usmerena na pokretanje, organizovanje i inoviranje poslovanja neke kompanije, sa osnovnim ciljem stvaranja novog tržišta i ostvarivanja dobiti.
- Vezano je blisko za aspekte ljudskog ponašanja: razvija kreativnost, pospešuje rađanje ideja i obogaćuje ljudske potrebe.
- Šta obuhvata?
Zbir znanja, veština i sposobnosti, ali i kreativnost, odgovornost, dinamičnost, posvećenost, upornost,...



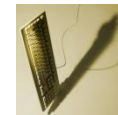
Finansijeri

- Ko finansira projekte?
 - države i ministarstva
 - NVO
 - velike kompanije
 - *Venture Capital*



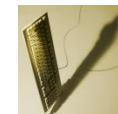
Startapovi i preduzetništvo

- Startap = pokreni kompaniju, ako imaš ideju (najčešće početnici u poslovanju koji traže mikro kredite za osnivanje svoje firme)
- Zašto odustajemo:
 - "Ne mogu to da uradim"
 - "To je neizvodljivo"
 - "Postoje već slične ideje, nemam šansu u tome"
 - "To su već probali pa nije uspelo"
- Postoje rizici - finansijski, društveni, psihološki
- Strah od rizika => Želja za uspehom
- Cilj: omogućiti brz povratak investicije i brz prodor/rast
- Dobra ideja i dobra motivacija!



Poznati startapovi u svetu

Naziv	Godina	Osnivači	# zaposleni	Brend?
HP	1939	Vilijam Hjulit, Dejvid Pakard	55 000	
Intel	1968	Endru Grouv, Gordon Mur, Robert Nojs	107 000	
Atari	1972	Nolan Bušnel	~ 1000	
Microsoft	1975	Bil Gejts, Pol Alen	134 900	
Apple	1976	Stiv Džobs, Stiv Voznijak	132 000	
Yahoo!	1994	Dejvid Filo, Džeri Jang	8 600	
eBay	1995	Pjer Omidar	34 600	
Google	1998	Lari Pejdz, Sergej Brin	98 770	



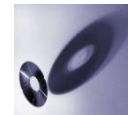
Poznati startapovi kod nas

Naziv	God.	Osnivači	# zaposleni	Grad
Nordeus	2010	Branko Milutinović, CEO Milan Jovović, CSO Ivan Stojsavljević, CTO	180	Beograd
Eipix Entertainment	2005	Mirko Topalski CEO Vladimir Živković CCO Adrian Đura CTO	350	Novi Sad
SevenBridges	2009	Igor Bogićević	250	Beograd
Frame	2012	Nikola Božinović	80	Niš / Beograd

Nordeus: Prihod od 12.66 miliona EUR u 2016. godini

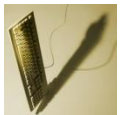
Seven Bridges dobio 45 miliona \$ u 2016. za razvojnu platformu.

Frame prodat kompaniji Nutanix za 190 miliona \$!



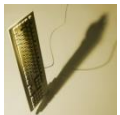
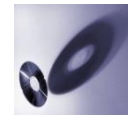
Sedam projektnih faktora

- Koje faktore treba uzeti u obzir prilikom strukturiranja softverskog tima?
 - težina problema koji treba rešiti
 - veličina rezultujućeg programa u linijama koda
 - vreme koje će tim provesti zajedno (team lifetime)
 - stepen u kojem problem može biti izdelfen na module
 - zahtevani kvalitet i pouzdanost sistema koji se pravi
 - rok završetka softvera (deadline)
 - komunikacija u timu



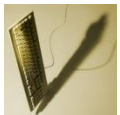
Ima i neuspešnih projekata

- Projekti vrlo često nisu uspešni
- SAD, 2003:
 - analizirano 13522 ICT projekata
 - zaključak: samo 1/3 projekata je bila uspešna
 - 82% projekata je kasnilo
 - 43% projekata je premašivalo predviđeni budžet
- Vrlo čest razlog pojavljivanja nedostataka u projektima je upravljanje projektima!



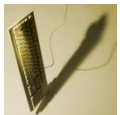
Osobine kompanija sa uspešnim projektima

- Izvršna podrška (top menadžment)
- Aktivno učešće korisnika u projektu
- Iskustvo menadžera projekta
- Jasni poslovni ciljevi
- Minimizovan obim
- Standardna softverska infrastruktura
- Čvrsti osnovni zahtevi
- Formalna metodologija
- Pouzdane procene



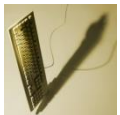
Kada je projekat ugrožen? (1)

- Deset znakova koji ukazuju da je projekat ugrožen:
 1. programeri ne razumeju potrebe korisnika
 2. obim aplikacije je slabo definisan
 3. promene se sprovode loše
 4. izabrana tehnologija se menja
 5. promena poslovnih potreba
(ili su loše definisane)



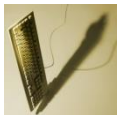
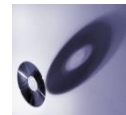
Kada je projekat ugrožen? (2)

6. rokovi su nerealni
 7. korisnici su otporni
 8. sponzorstvo je propalo
(ili nikad nije bilo dobijeno)
 9. timu nedostaju ljudi sa odgovarajućim znanjem
 10. menadžeri i stručnjaci izbegavaju praksu i učenje lekcija
- Kog tipa mogu biti greške?
 - U vezi sa ljudima, procesom, proizvodom, tehnologijom



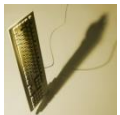
Misija i vizija kompanije/organizacije

- Misija odgovara na pitanje zašto postojimo.
- Vizija odgovara na pitanje gde želimo da stignemo - naš dugoročan, racionalan i pomalo emotivan cilj.
- Ciljevi nam kažu gde treba da idemo i kako da stignemo.
- Strategija je način da ostvarimo cilj.



Menadžment

- “In-house” projekti
- Sve više organizacija nema ICT sektore
- Menadžer projekta, sa klijentske strane, zadužen za nadzor ugovora
- Menadžer projekta, sa strane proizvođača softvera, bavi se više tehničkim pitanjima
- Mnoge tehničke odluke delegira softverskim inženjerima



Tri uzastopna procesa za razvoj novog sistema

Studija izvodljivosti

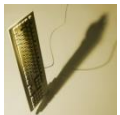
Da li je vredno
truda?

Kako ćemo
uraditi projekat?

Plan

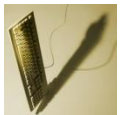
Izvršavanje projekta

Rad, rad, rad



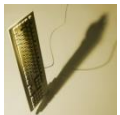
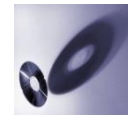
Studija izvodljivosti

- Studija izvodljivosti je važan korak u bilo kom procesu razvoja softvera.
- Studija predstavlja analizu različitih aspekata kao što su troškovi potrebni za razvoj i izvršavanje sistema, vreme potrebno za svaku fazu sistema i tako dalje.
- Ako ovi važni faktori nisu analizirani onda to ima uticaj na organizaciju i razvoj i sistem će biti neuspešan.
- Za pokretanje projekta i organizacije, ovaj korak je veoma važan korak u procesu razvoja softvera!



Prednosti studije izvodljivosti

- Učestvuje u analizi
 - Pomaže u identifikovanju faktora rizika i u planiranju za analizu rizika
 - Pomaže u izradi analize troškova koja je vrlo korisna za organizaciju
 - Pravi planove za obuku programera koji će raditi implementaciju sistema
-
- Studija izvodljivosti = Izveštaj koji se koristi od strane top menadžmenta (ljudi) u organizaciji. Na osnovu tog izveštaja se odlučuje o proceni troškova, finansiranju i drugim važnim odlukama za organizaciju.



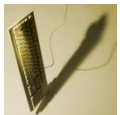
Planiranje projekta

- Ukoliko studija izvodljivosti pokaže da je potencijal projekta dobar, tada planiranje projekta može da počne.
- Obuhvata:
 - Određivanje zahteva
 - Određivanje resursa
 - Izbor modela životnog ciklusa
 - Odrediti karakteristike proizvodne strategije
- Kod velikih projekata, ne treba da pravimo detaljno planiranje na samom početku.
- 2-3 osobe, oko 6 meseci - mali ili veliki projekat?
- Mali projekat može biti deo velikog, ali samo ukoliko posmatramo mali projekat kao jedinstvenu celinu.
- Planiranje daljih faza se ostavlja da bude bliže njihovom početku, kada znamo više detalja i informacija o ranijim fazama.



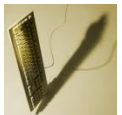
Izvršavanje projekta

- Čine ga 2 podfaze (najčešće):
 - dizajn
 - implementacija
- Dizajn predstavlja odlučivanje o formi proizvoda koji se pravi (na primer: spoljašnji izgled softvera - korisnički interfejs ili unutrašnja arhitektura sistema)
- Plan opisuje aktivnosti koje treba da budu realizovane prilikom izvršavanja projekta
- Planiranje vrlo često može biti pod uticajem odluka o dizajnu



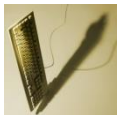
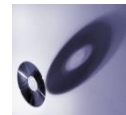
Praćenje projekta

- Šta pratimo?
 - Troškove
 - Napor
 - Raspored
- Planirano protiv aktuelnog
- Kako se upravljamo kada stvari ne idu svojim tokom? (idu van plana)



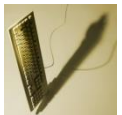
Merljivost projekta

- Može da se meri ono što je do danas urađeno na projektu:
 - trošak (u plusu, minusu ili na pozitivnoj 0)
 - napor (broj uložениh resursa)
 - raspored (da li faze/aktivnosti ne kasne)
- Alternative
- Trenutna vrednost (zarada)
- Produktivnost (npr. broj linija koda)
- Složenost (npr. broj funkcionalnih poena)



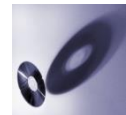
Tehničke osnove

- Zahtevi
- Analiza
- Dizajn sistema
- Konstrukcija
- Osiguranje kvaliteta
- Razvoj



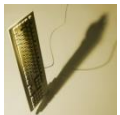
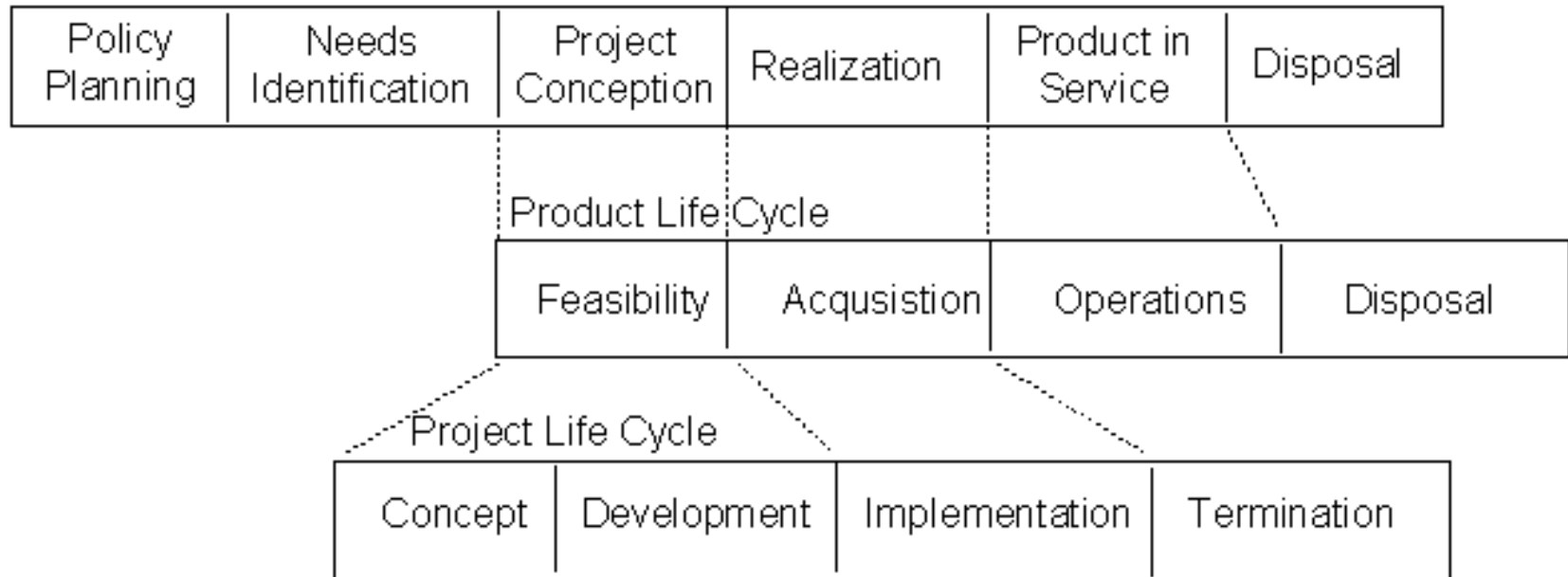
Projektne faze

- Svi projekti treba da budu podeljeni u faze
- Sve faze zajedno čine životni ciklus projekta
- Svaka faza treba da ima svoj(e) rezultat(e) (*deliverables*)
- Potrebno je identifikovati glavne (primarne) faze softverskog projekta

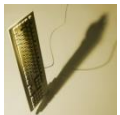
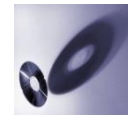
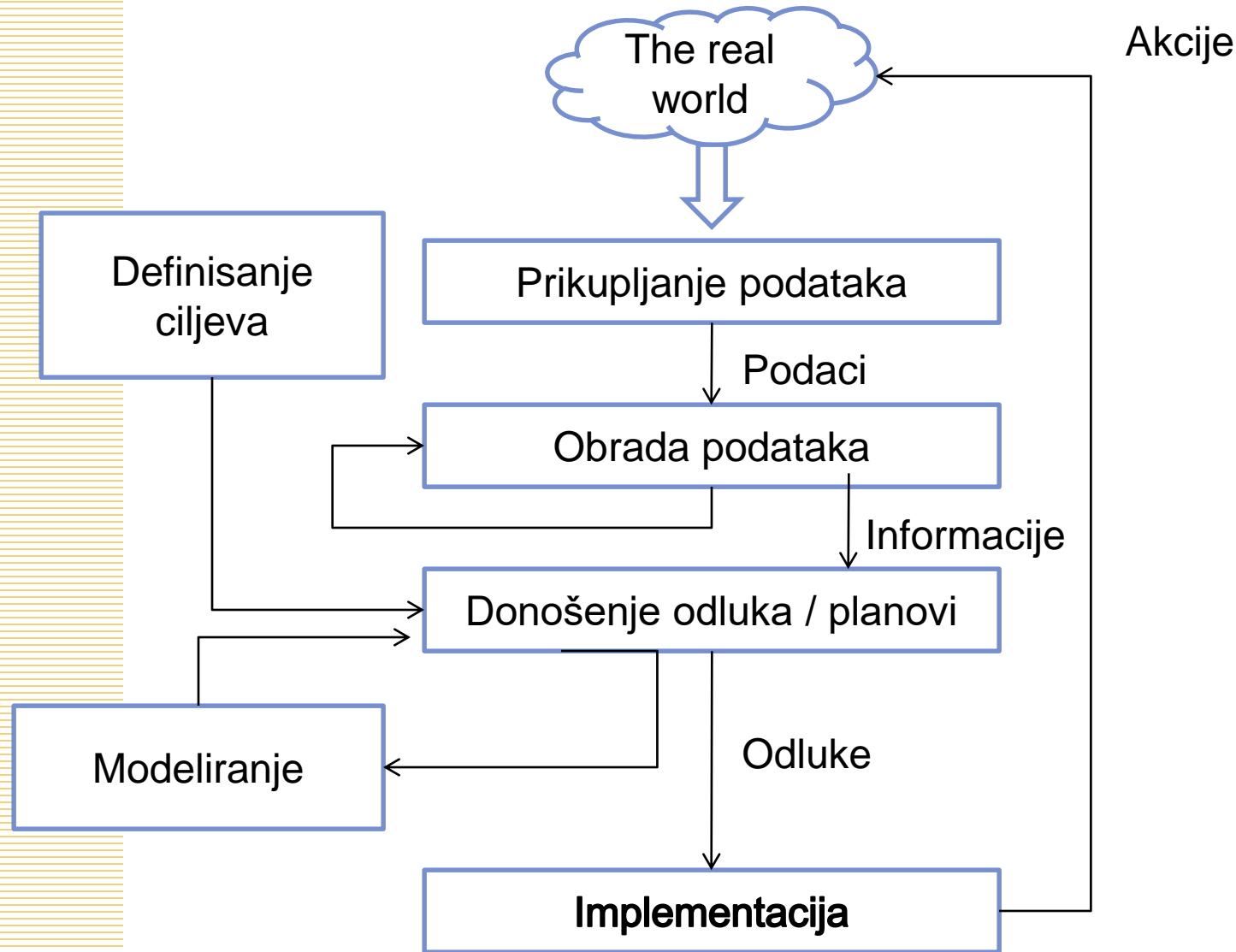


Veze između životnih ciklusa

Business Life Cycle

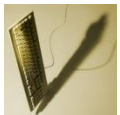


Ciklus upravljanja projektima



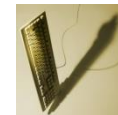
Projektni plan

- Plan treba konstantno prilagođavati tokom izvođenja projekta i kaže se da je on zbog toga **dinamički**.
- Na skoro svim projektima, više vremena se provede radeći projekat, nego u planiranju.
- Dobar plan obezbeđuje osnovu za dobar projekat!



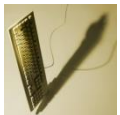
Sadržaj projektnog plana

- Uvod
- Pozadina: uključuje referencu na studiju slučaja
- Projekti ciljevi
- Ograničenja
- Metode
- Produkt(i) projekta
- Aktivnosti koje će biti sprovedene
- Korišćeni resursi
- Projektni rizici
- Upravljanje projektom, obuhvata organizaciju, upravljanje kvalitetom, upravljanje rasporedom



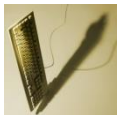
Studija slučaja (Business case)

- Organizacije mogu imati različita dokumenta - studije izvodljivosti (a feasibility study) ili opravdanost projekta (a project justification), za ono što ćemo mi nazivati studija slučaja.
- Cilj ovog dokumenta je da obrazloži projekat, i da prikaže da će prednosti rezultata projekta premašiti troškove razvoja, implementacije i rada.



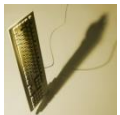
Studija slučaja (Business case)

- Obično dokument studija slučaja može da sadrži:
 - Uvod i pozadinu predloga
 - Predloženi projekat
 - Tržište
 - Organizacionu i radnu infrastrukturu
 - Prednosti
 - Nacrt plana implementacije
 - Troškove
 - Finansijski slučaj
 - Rizike
 - Menadžment plan (plan upravljanja projektom)



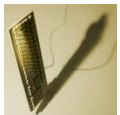
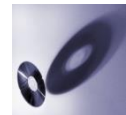
Studija slučaja (Business case)

- Uvod i pozadina
Opis trenutnog okruženja predloženog projekta. Opis problema koji treba rešiti. Opis nekih pogodnosti koje su identifikovane za proces razvoja projekta.
- Predloženi projekat
Navesti kratak pregled predloženog projekta.
- Tržište
Ovo je jako bitno ispitati kada je rezultat projekta novi proizvod ili novi servis. Treba analizirati potražnju za proizvodom/servisom i analizirati konkurente proizvode.



Studija slučaja (Business case)

- Organizaciona i radna infrastruktura
Opisuje kako će struktura organizacije uticati na realizaciju projekta. Od velike važnosti je i da li se projekat implementira ili modifikuje informacioni sistem, kao deo šireg projekta.
- Prednosti
Gde je to moguće, finansijska vrednost treba da bude stavljena u prvi plan za realizovani projekat. U komercijalnim organizacijama, bitno je povećati prihode ili smanjiti troškove. Kod neprofitnih organizacija, bitno je klasifikovati prednosti, čak i ako ne može da se odrede finansijske vrednosti.



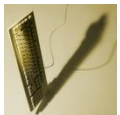
Studija slučaja (Business case)

- Nacrt plana implementacije

Pored ICT aspekata projekta, aktivnosti kao što su marketing, promocije, radna infrastruktura i održavanje moraju se uzeti u obzir.

Jedno razmatranje mogu biti aktivnosti koje će biti outsource-ovane, a koje aktivnosti je bolje da ostanu “u kući”.

Ovde će se takođe detaljno utvrditi upravljanje implementacijom. Dodeljuju se dužnosti za zadatke koji su identifikovani. Treba identifikovati i ključne tačke/prekretnice (milestones).



Studija slučaja (Business case)

- Troškovi

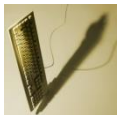
Pošto smo u predlogu istakli korake potrebne za realizaciju projekta, može se predstaviti i raspored očekivanih osnovnih troškova.

Detaljnija analiza troškova je neizvesna ukoliko nemamo detaljnijih zahteva.

- Finansijski slučaj

Postoji više načina preko kojih možemo da analiziramo prihode i troškove.

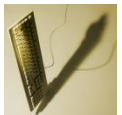
(biće kasnije reći o tome)



Studija slučaja (Business case)

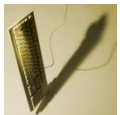
- Rizici

Kao i kod ciljeva, možemo utvrditi razliku između projektnih rizika (rizici koji se odnose na pretnje za uspešan završetak projekta) i poslovnih rizika (koji se odnose na neke faktore, uz praćenje uspešnosti isporučenog softvera).



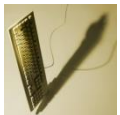
Evaluacija pojedinačnih projekata

- Tehnička procena predloženog sistema sastoji se od procene da li se zahtevana funkcionalnost može postići sa aktuelnim tehnologijama.
- Organizaciona politika, čiji je jedan od ciljeva i pružanje stalne softversko-hardverske infrastrukture, vrlo često može da ograniči tehnička rešenja.
- Troškovi odabranih tehnologija moraju biti uzeti u obzir u analizi troškova i dobiti.



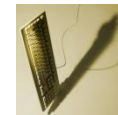
Analiza troškova i dobiti

- Čak i tamo gde je procenjena dobit veća od procenjenih troškova, često treba odlučiti da li je predloženi projekat najbolja varijanta od nekoliko opcija. Ne mogu se svi projekti izvršiti u svakom trenutku i u svakom slučaju, najveći projekti traže najveći broj resursa.
- Analiza troškova i dobiti obuhvata 2 koraka:
- a) Identifikovanje svih troškova i dobiti od urađenog projekta i rada isporučene aplikacije.
- b) Izraziti troškove i dobiti u istim jedinicama.



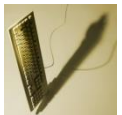
Podela troškova

- Većina direktnih troškova su već izraženi u monetarnim jedinicama i mogu biti grupisani u sledeće kategorije:
- troškovi razvoja
- troškovi instalacije
- troškovi rada



Neto profit (dobit)

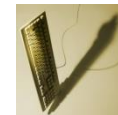
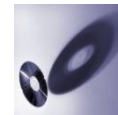
- Neto profit (dobit) je ukupna razlika između svih prihoda i svih rashoda tokom trajanja projekta.



Projekcija neto profita

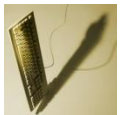
GODINA	Projekat 1	Projekat 2	Projekat 3	Projekat 4
0	-100 000	-1 000 000	-100 000	-120 000
1	10 000	200 000	30 000	30 000
2	10 000	200 000	30 000	30 000
3	10 000	200 000	30 000	30 000
4	20 000	200 000	30 000	30 000
5	100 000	300 000	30 000	75 000
Neto profit	50 000	100 000	50 000	75 000

Projekcija profita može da se radi i na kvartalnom (3 meseca) ili mesečnom nivou.



Period otplate

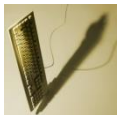
- Period otplate je vreme potrebno da se vrate investicije koje su uložene na početku.
- Kompanije uvek biraju projekat sa najkraćim rokom otplate, jer žele da smanje vreme za koje je projekat u dugovima.
- Koje su prednosti, a koji su nedostaci?



Povračaj investicije

- Return on investment (ROI) ili Accounting rate of return (ARR) obezbeđuje način da uporedimo neto profitabilnost i zahtevano investiranje.

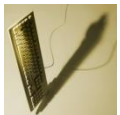
$$ROI = \frac{\textit{godišnja_dobit_prosek}}{\textit{ukupne_investicije}} \times 100$$



Neto sadašnja vrednost - NSV

- Obračun neto sadašnje vrednosti je tehnika koja uzima u obzir isplativost projekta u odnosu na vreme.

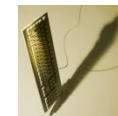
$$Sadašnja_vrednost_NSV = \frac{vrednost_u_godini_t}{(1+r)^t}$$



Izračunavanje NSV

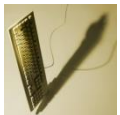
Godina	Projekat 1 (EUR)	Stopa @10%	Obračunata vrednost (EUR)
0	-100 000	1.0000	-100 000
1	10 000	0.9091	9 091
2	10 000	0.8264	8 264
3	10 000	0.7513	7 513
4	20 000	0.6830	13 660
5	100 000	0.6209	62 090
Neto profit:	50 000 EUR		NPV: 618 EUR

Glavna poteškoća kod NSV je izbor odgovarajuće stope. Neke organizacije imaju standardnu stopu, ali tamo gde to nije slučaj, onda stopa treba da bude izabrana tako da odražava prosečnu kamatnu stopu + premije koje predstavljaju činjenicu da su softverski projekti više rizični nego ulaganje novca u banku.



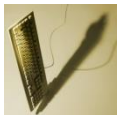
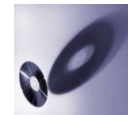
Interna stopa povraćaja - IRR

- *Internal rate of return*
- Pokušava da izračuna profitabilnost kao procenat povraćaja
- Microsoft Excel podržava ovu funkciju (IRR)
- Da li je bolji:
 - projekat sa NSV 100 000 i IRR 15%
 - projekat sa NSV 10 000 i IRR 18%



Procena rizika

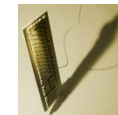
- Svaki projekat mora da obuhvati rizike!
- Projektni rizici - oni koji sprečavaju da se projekat uspešno završi
- Poslovni rizici - isporučeni softver nije isplativ



Identifikacija i kategorizacija rizika

Rizik	Značaj	Verovatnoća
Klijent odbacuje predloženi izgled sajta	H	-
Konkurenti spuštaju cene	H	M
Skladišta ne mogu da se izbore sa povećanom potražnjom	M	L
Online plaćanja preko neke banke, imaju bezbedonosne probleme	M	M
Troškovi održavanja su veći od procenjenih	L	L
Vreme odziva je sporo, pa kupci nisu zadovoljni	M	M

H = veliki rizik, M = srednji rizik, L = mali rizik
- = izuzetno malo verovatno



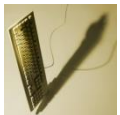
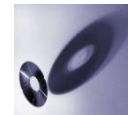
Planiranje projekta - Agenda

1. Prince2 metodologija upravljanja

2. Glavni dijagrami u upravljanju soft. projektima

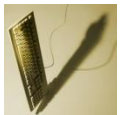
3. Modeli životnog ciklusa softvera

4.

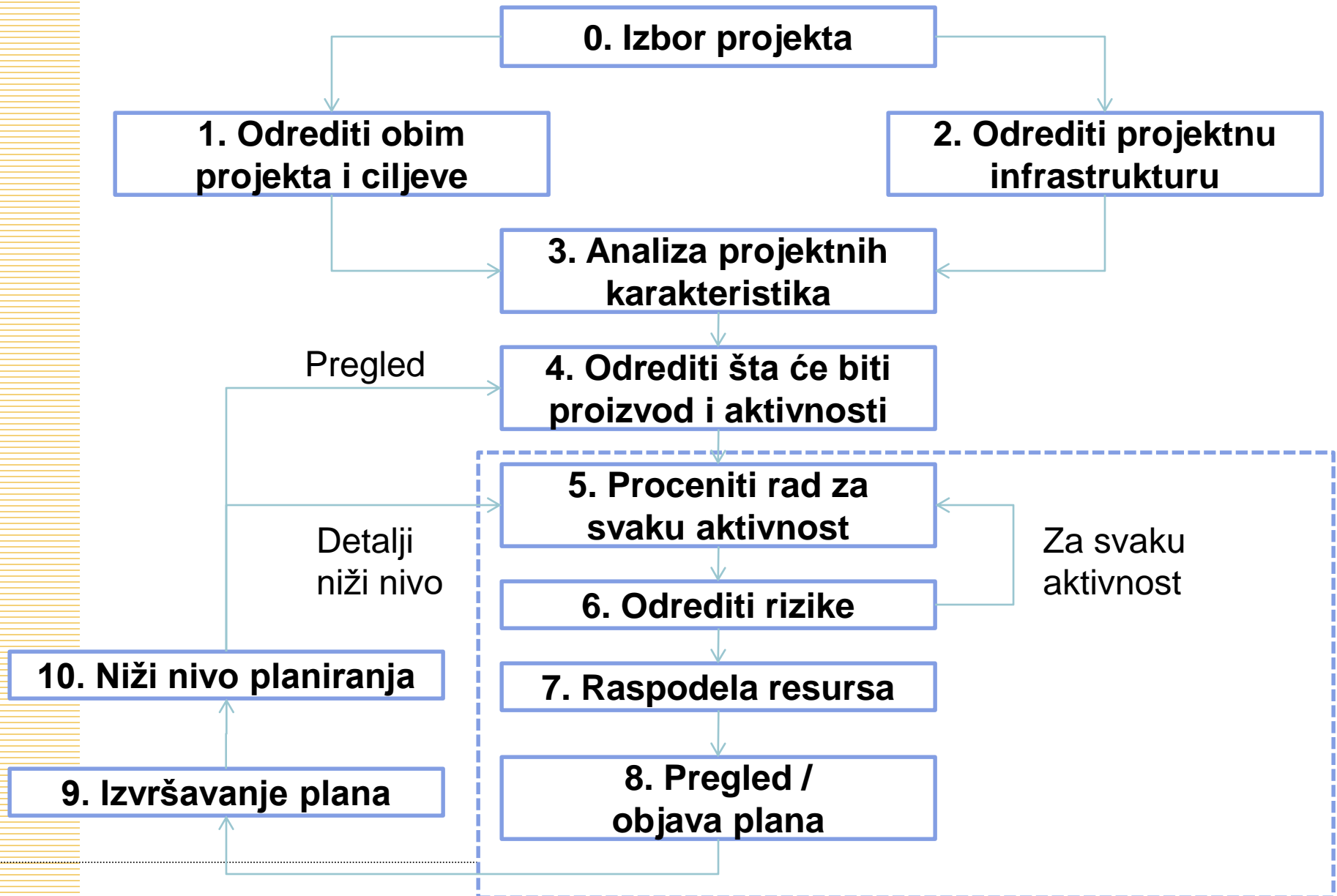


Prince2 i SWPMM

- **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.

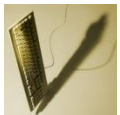


Step Wise PMM - pregled



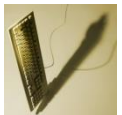
Korak 0: Izbor projekta

- 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



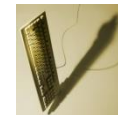
Korak 2: Odrediti projektnu infrastrukturu

- 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

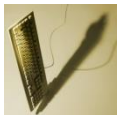
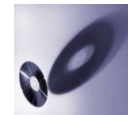
2.2: Odrediti standarde i procedure

2.3: Odrediti organizaciju projektnog tima



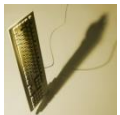
Korak 3: Analiza projektnih karakteristika

- 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



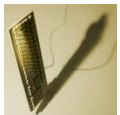
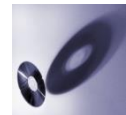
Korak 4: Odrediti šta će biti proizvod i aktivnosti

- 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



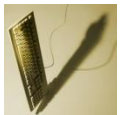
Identifikovati konačni proizvod

- 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)



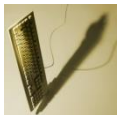
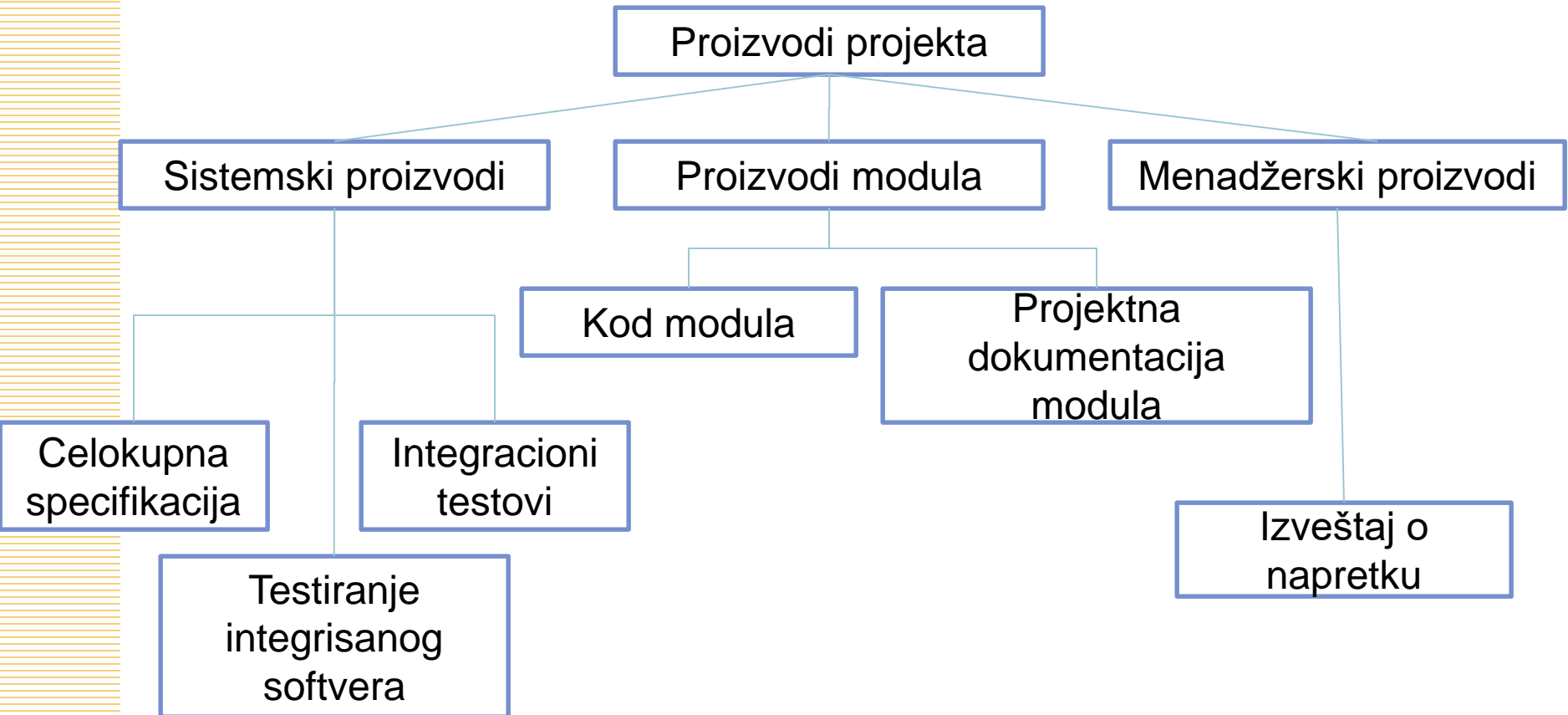
Stablo aktivnosti (rada) - WBS

- *Work Breakdown Structure (WBS)*
- *Primer (tekstualni):*
 - **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**



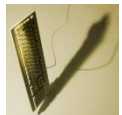
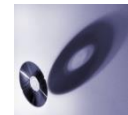
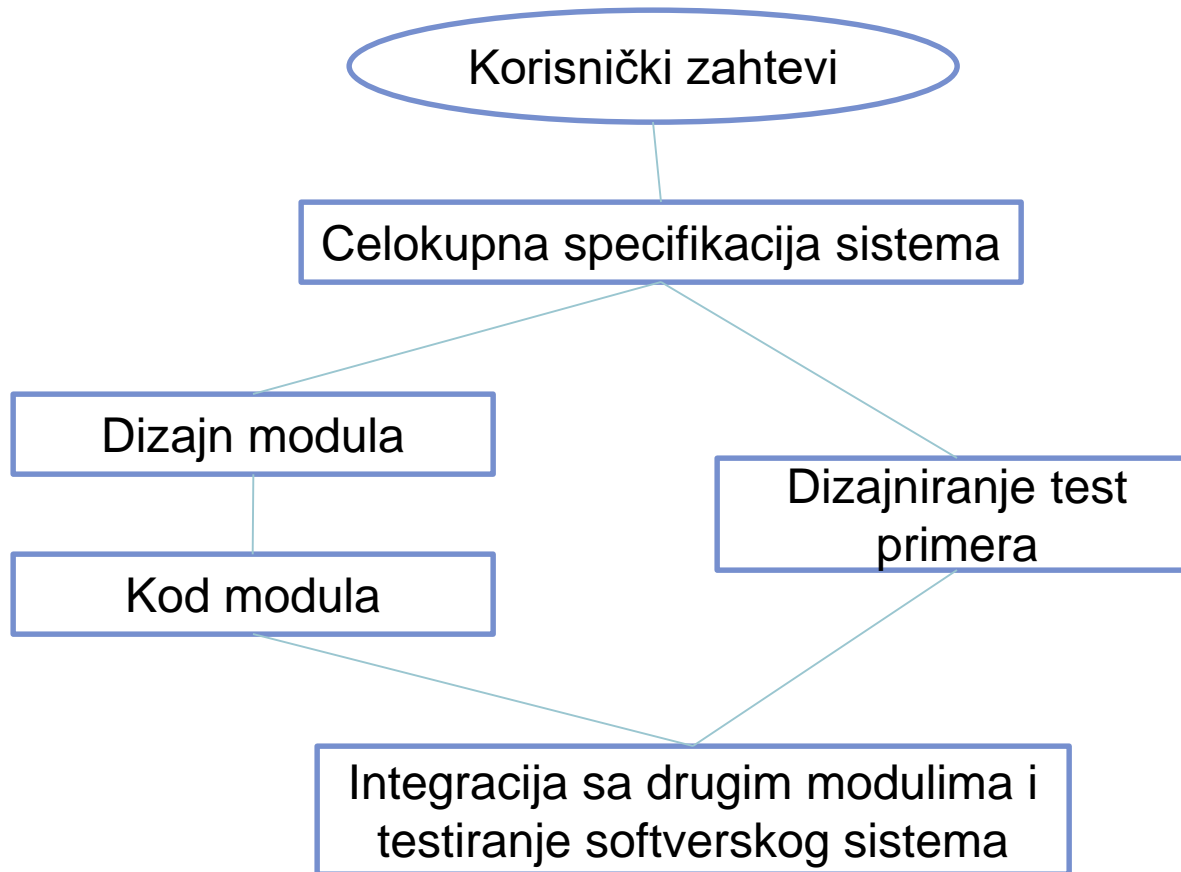
Stablo proizvoda - PBS

- *Product Breakdown Structure (PBS)*



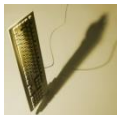
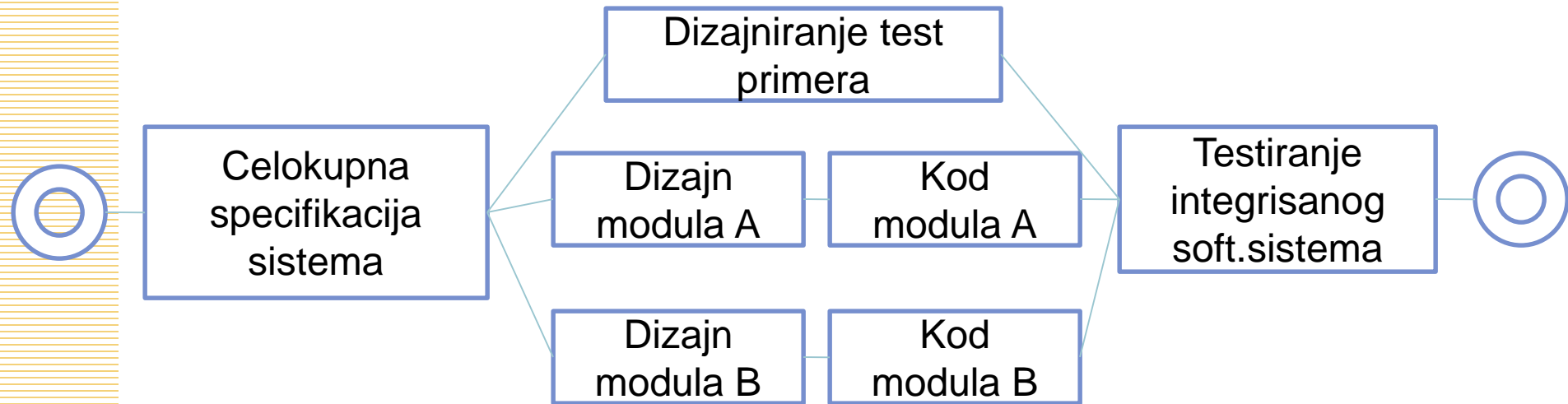
Dijagram toka proizvoda - PFD

- *Product Flow Diagram (PFD)*



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.



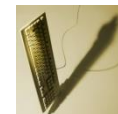
Korak 5: Proceniti rad za svaku aktivnost

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.



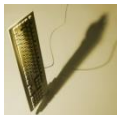
Korak 5: Proceniti rad za svaku aktivnost

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

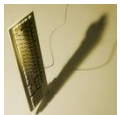
6.1: Identifikuje i broji rizike zasnovane na aktivnostima

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

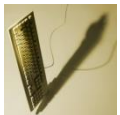
7.1: Utvrditi i rasporediti resurse

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!



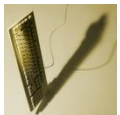
Korak 8: Pregled / objava plana

8.1: Razmatranje kvaliteta

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

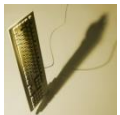
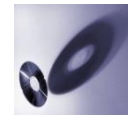
8.2: Dokumentovati plan i dobiti saglasnost

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.



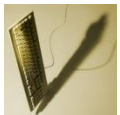
Koraci 9 i 10: Izvršavanje plana i niži nivo planiranja

- 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.



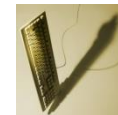
Izbor metodologija i tehnologija (1)

- 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.



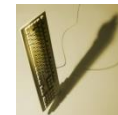
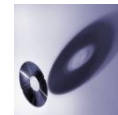
Izbor metodologija i tehnologija (2)

- 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

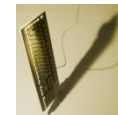
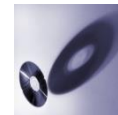


Analiza projektnih karakteristika

- 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?

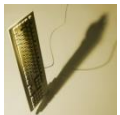


PLANIRANJE AKTIVNOSTI KORIŠĆENJEM DIJAGRAMA



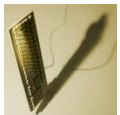
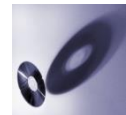
Organizacija projektnih aktivnosti

- 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 milestone-ova je pravolinijska (krajevi pojedinih faza).

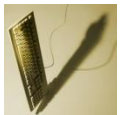
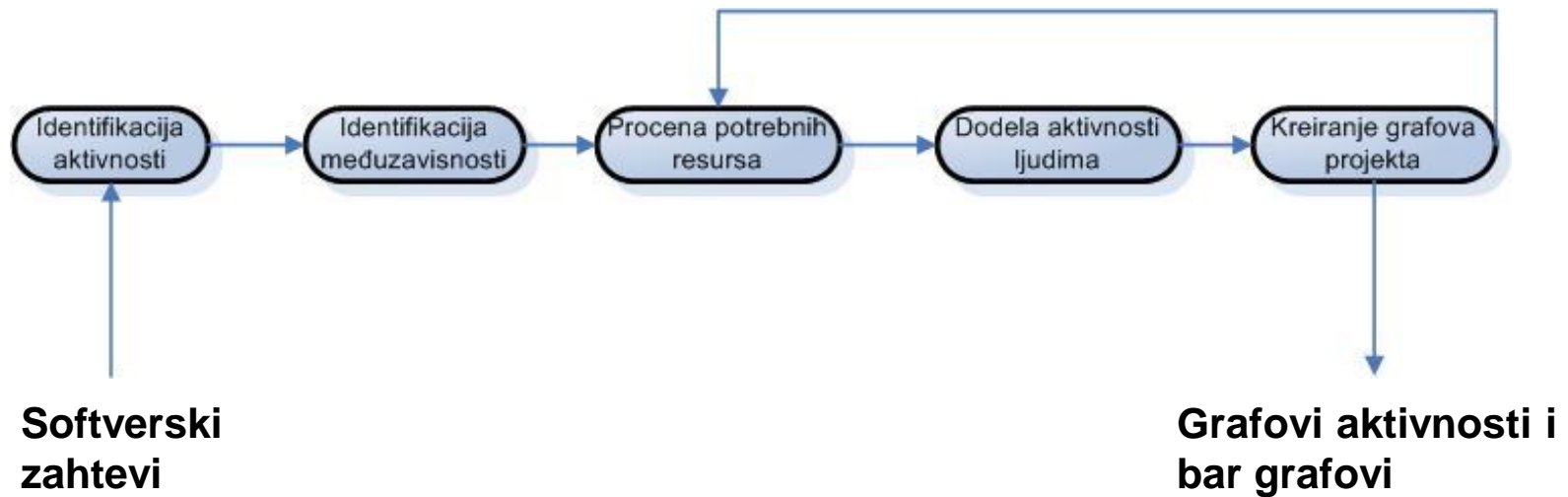


Raspoređivanje projektnih aktivnosti

- Izdeliti projekat u zadatke/poslove (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

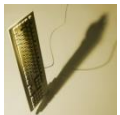
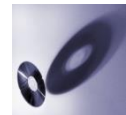


Proces raspoređivanja aktivnosti



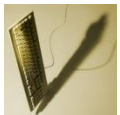
Problemi u raspoređivanju aktivnosti

- 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.



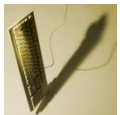
Grafičko predstavljanje rasporeda aktivnosti

- 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
- Tipovi u softverskim projektima:
 - Mreža aktivnosti metodom kritične putanje (CPM)
 - Gantov dijagram (bar graf vremena i aktivnosti)
 - PERT dijagram



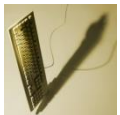
Model mreže aktivnosti

- 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!



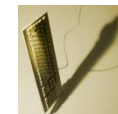
Labele u mreži aktivnosti

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



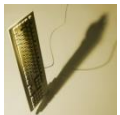
Primer projektne specifikacije

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



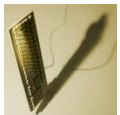
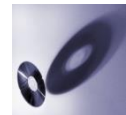
Metod kritične putanje (Critical Path method, CPM)

- 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)

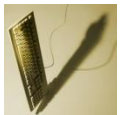
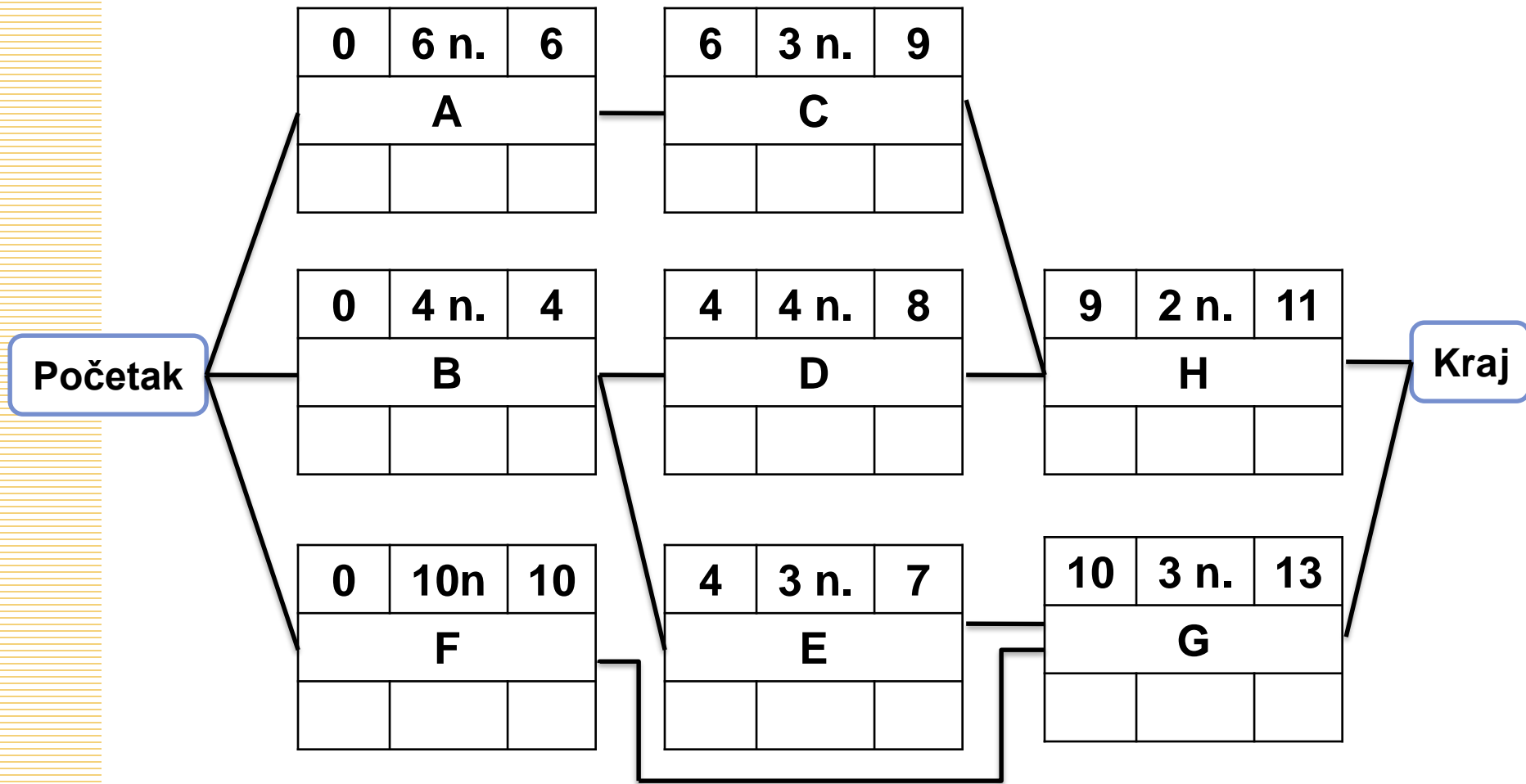


Prvi prolaz CPM

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

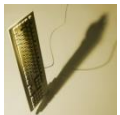
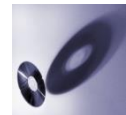


Prvi prolaz - prolaz unapred

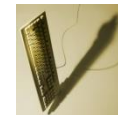
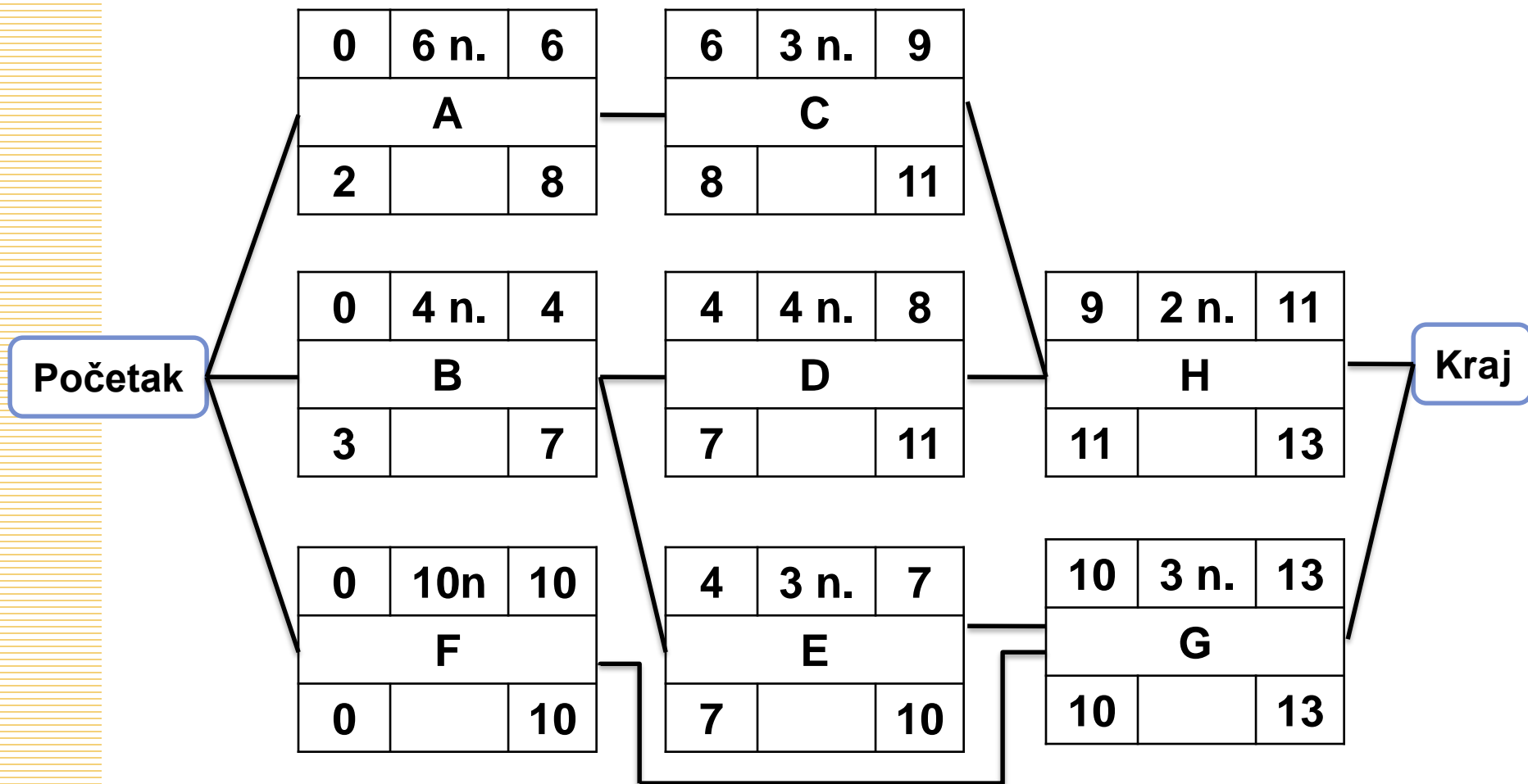


Drugi prolaz CPM

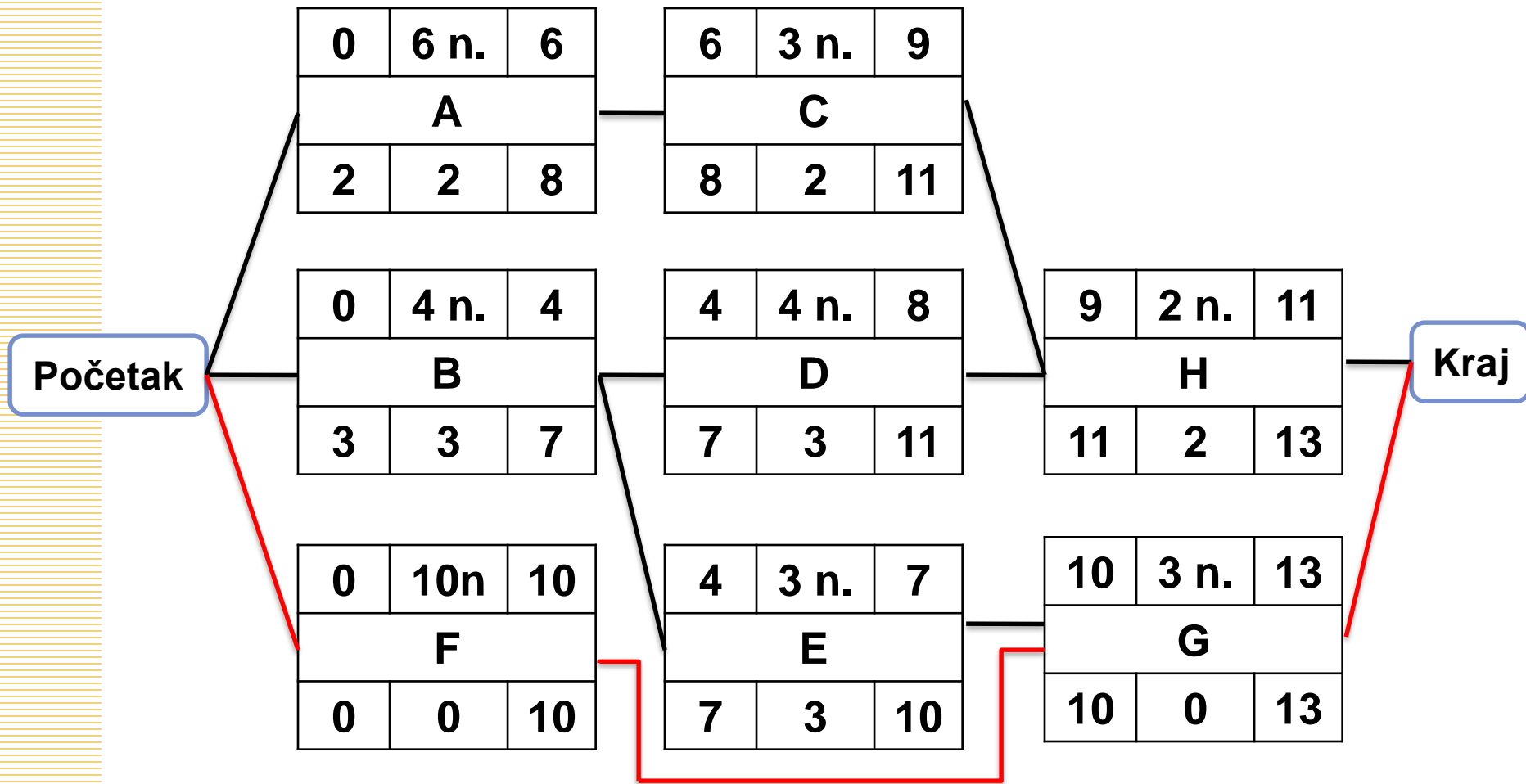
- Prolazi se kroz graf unazad počev do završnog čvora (čvorova) ka startnim
- Za svaku aktivnost odredi se Late Start Time (LST) i Late Finish Time (LFT)
- LFT je jednako najmanjem LST svih aktivnosti sledbenika u grafu (za završni čvor, jednako je njegovom EFT)
- $LST = LFT - \text{vreme trajanja aktivnosti}$



Drugi prolaz - prolaz unazad

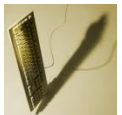


Kritični put



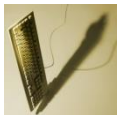
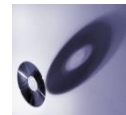
CPM

- 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)

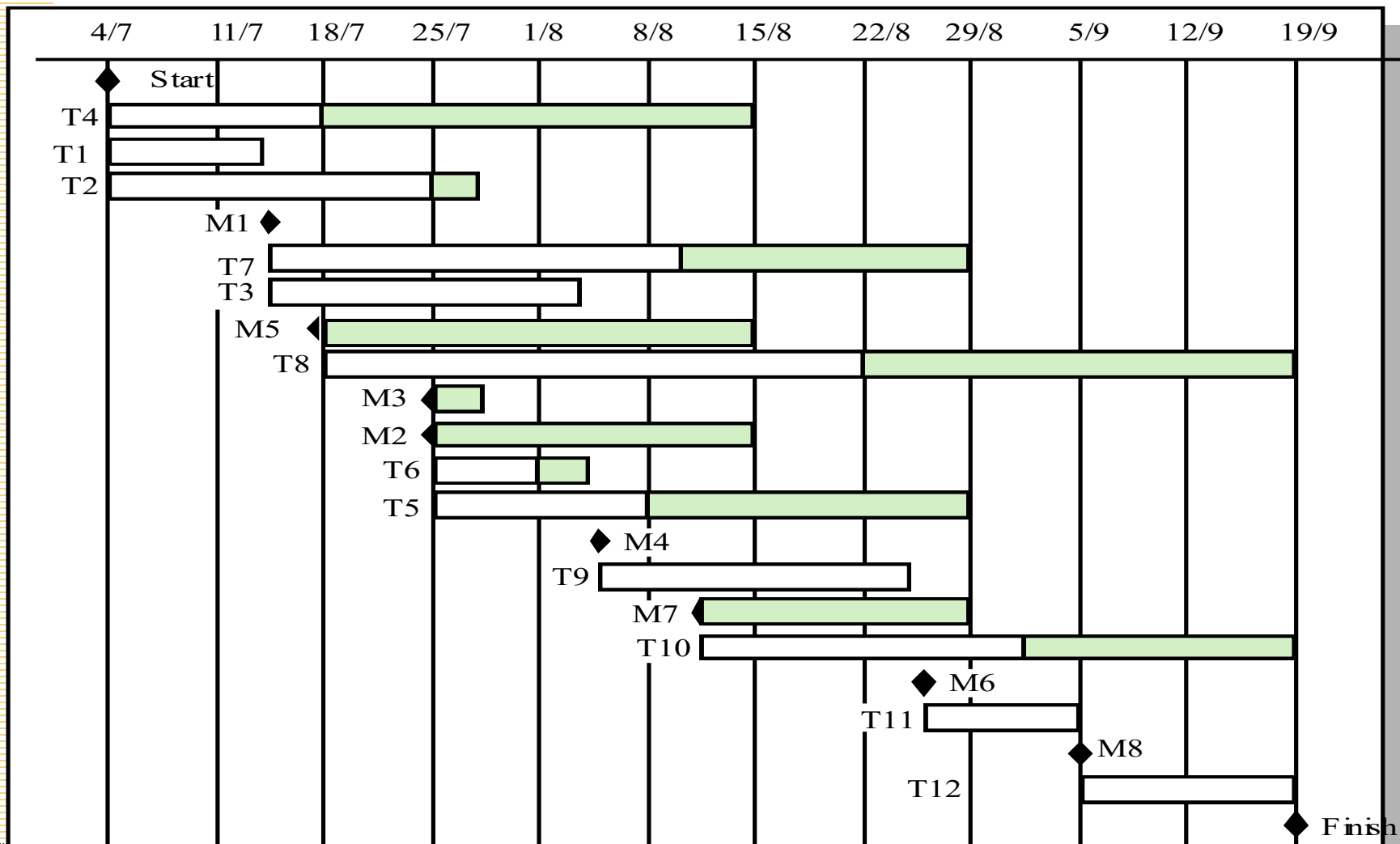


Vremenska linija aktivnosti (Gantov dijagram)

- Izumitelj *Henry Gantt* (1861–1919), oko 1910-1915.
- Vrsta bar dijagrama koji ilustruje raspored aktivnosti nekog projekta
- Moderne verzije ovog dijagrama prikazuju zavisnosti između aktivnosti, tačke prekretnice projekta, itd.
- Po horizontalnoj osi (x-osi) je vreme projekta, po vertikalnoj osi (y-osi) su aktivnosti (zadaci) projekta
- Gantov dijagram treba da prikaže za sve aktivnosti datume početka i datume kraja aktivnosti, bez obzira da li su terminalni ili sumarni zadaci, tako da na kraju znamo i datum početka i kraja za čitav projekat
- Sve aktivnosti prikazane u okviru ovog dijagrama čine stablo rada na projektu (WBS)

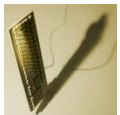


Vremenska linija aktivnosti (Gantov dijagram)



PERT tehnika

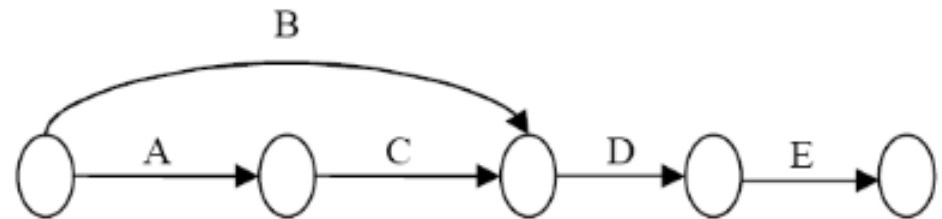
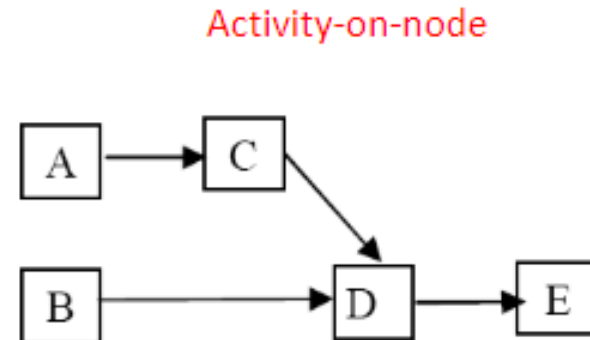
- *Program Evaluation and Review Technique*
- Tehnika evaluacija i pregleda programa/projekta, služi za analizu i predstavljanje zadataka koji moraju da se izvrše da bi projekat završili.
- PERT nudi alat za upravljanje koji se oslanja na dijagrame aktivnosti i događaje sa strelicama i čvorovima: strelice predstavljaju aktivnosti ili rad neophodan za dostizanje događaja (čvora), koji treba da označavaju svaku završenu fazu ukupnog projekta.
- PERT i CPM su slične, ali komplementarne tehnike:
 - CPM koristi samo jednu procenu vremena, i jednu procenu troškova za svaku aktivnost, a PERT koristi 3 vremenske procene i bez procene troškova za svaku aktivnost



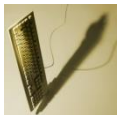
PERT i CPM diagram

Example of PERT diagrams:

Task	Precedence
A	
B	
C	A
D	B,C
E	D



Activity-on-arrow



PERT tehnika

- PERT se sve više primenjuje na raspoređivanje svih kritičnih putanja
- 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)

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

- Ovo vreme može da se primeni i kod Gantovog dijagrama, ako je tako naglašeno u procesu razvoja projektnog plana



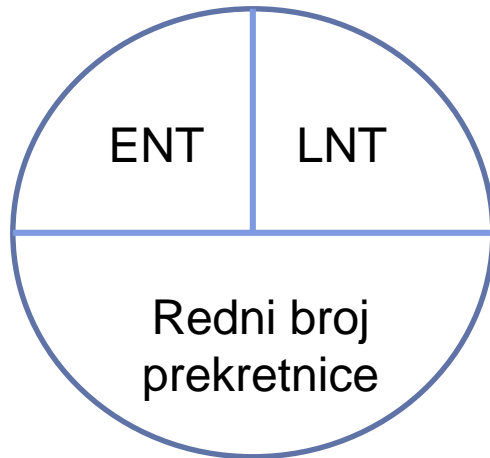
Primer za računanje vremena

Aktivnost (zadatak)	Preduslov	Obračun vremena			Očekivano vreme (T_E)
		Optim. (O)	Najvero- vatnije (NV)	Pesim. (P)	
<i>A</i>	/	2	4	6	4.00
<i>B</i>	/	3	5	9	5.33
<i>C</i>	<i>A</i>	4	5	7	5.17
<i>D</i>	<i>A</i>	4	6	10	6.33
<i>E</i>	<i>B, C</i>	4	5	7	5.17
<i>F</i>	<i>D</i>	3	4	8	4.50
<i>G</i>	<i>E</i>	3	5	8	5.17



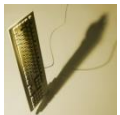
PERT tehnika

- Najčešće labele za čvorove (prekretnice) PERT dijagrama:

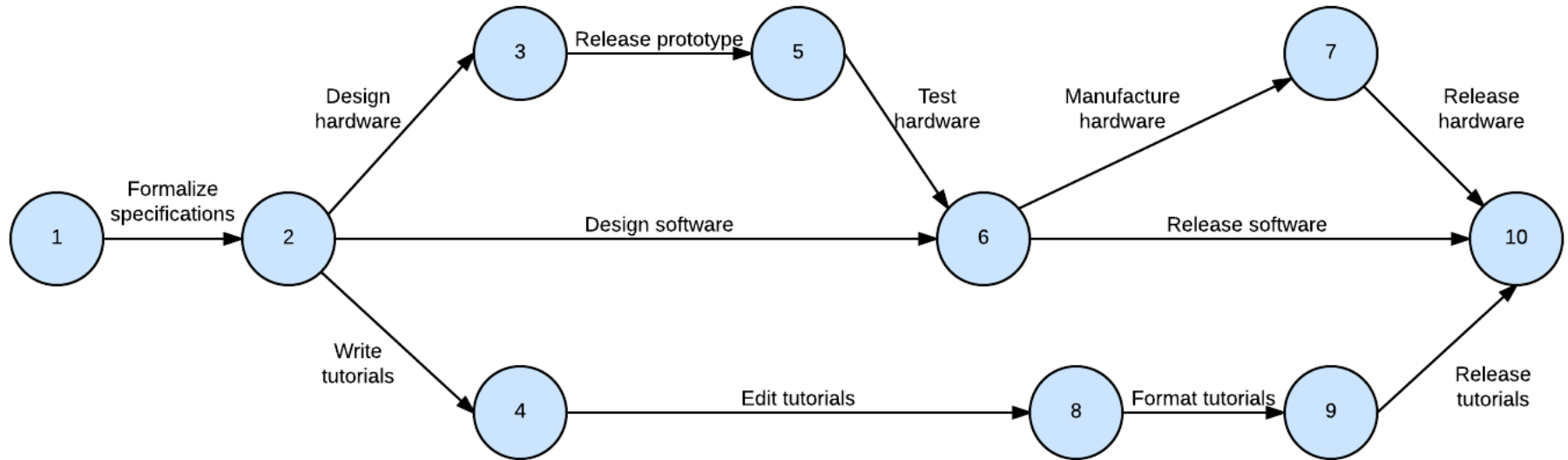


Redni broj prekretnice	Ciljni dan
Očekivani dan	Standardna devijacija

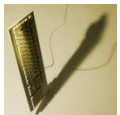
- ENT - najranije vreme dostizanja pretkrenice,
- LNT - najkasnije vreme dostizanja prekr. (bez kašnjenja)
- Ciljni dan kada će se dostići prekretnica najkasnije (LNT)
- Očekivani dan kada će biti dostignuta prekretnica (sa T_e)
- Standardna devijacija: $\sigma_{T_e} = (P - O) \div 6$



Primer PERT dijagrama (1)

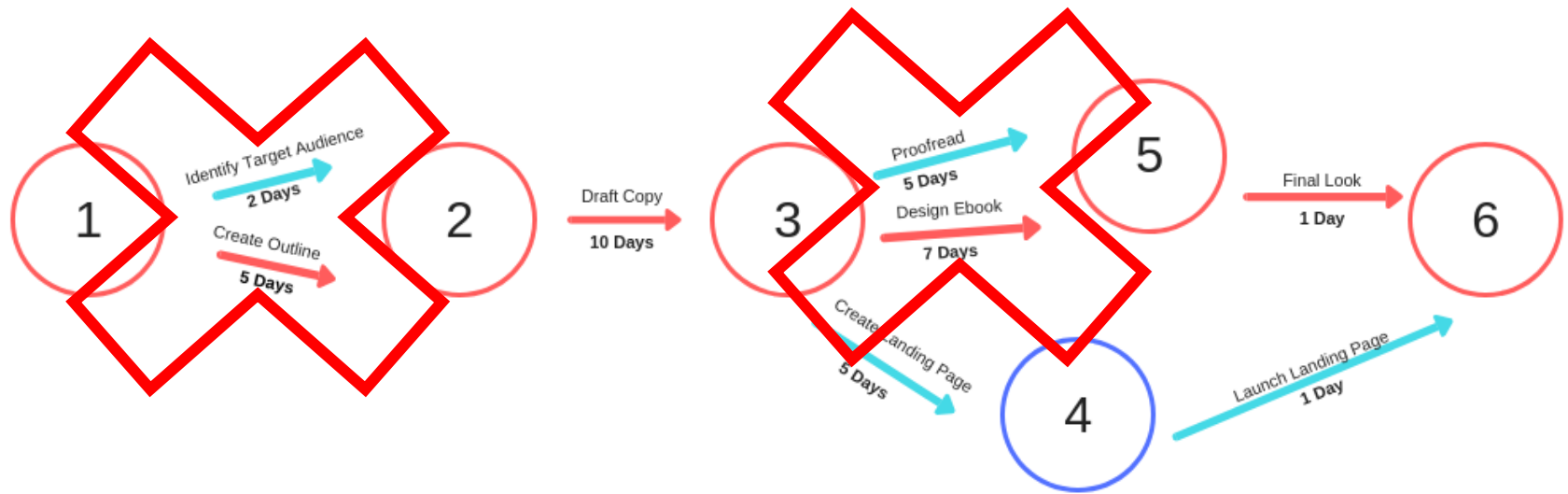


- Jedan početni i jedan krajnji čvor!
- Čvor je sada trenutak / prekretnica projekta
- Strelica je posao/aktivnost (iz WBS) i ima svoje trajanje

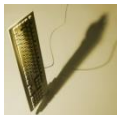


Primer PERT dijagrama (2)

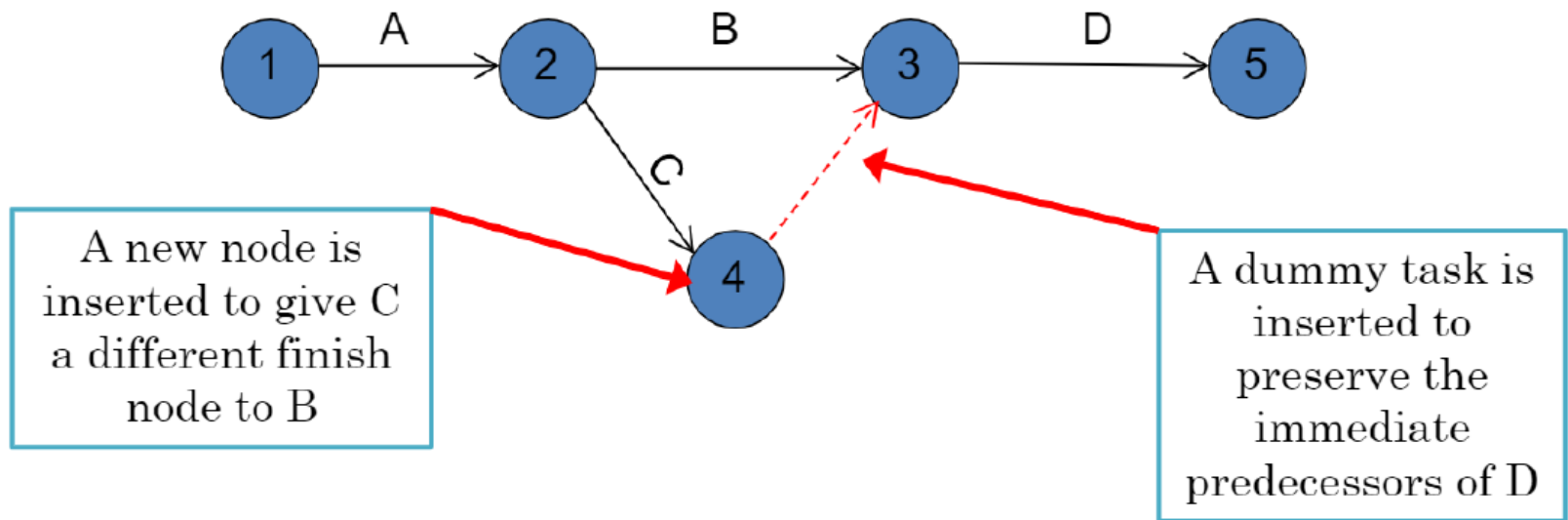
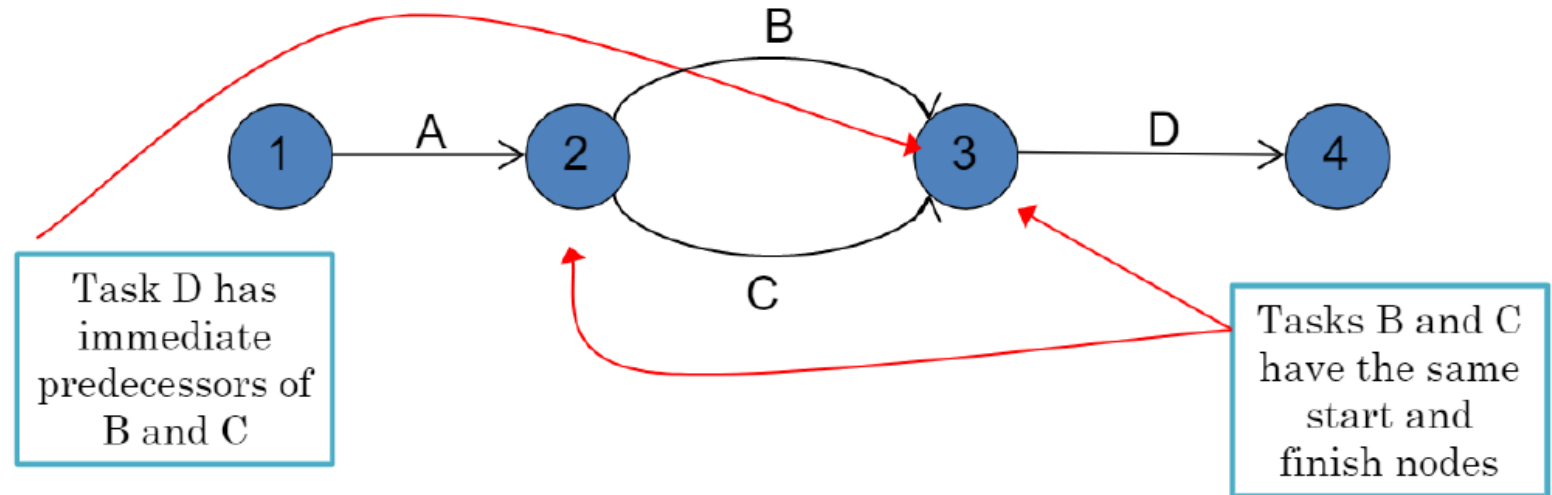
- Mogu da budu iz jednog čvora i dva ili više poslova/aktivnosti (dve ili više strelica) sa različitim trajanjem, ali ne mogu onda završiti u istoj prekretnici!



Kako rešiti ovakav problem?



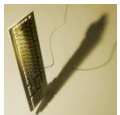
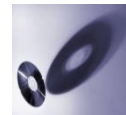
Veštačka aktivnost (dummy)



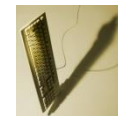
Aktivnosti u projektnom planu

- 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!

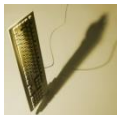
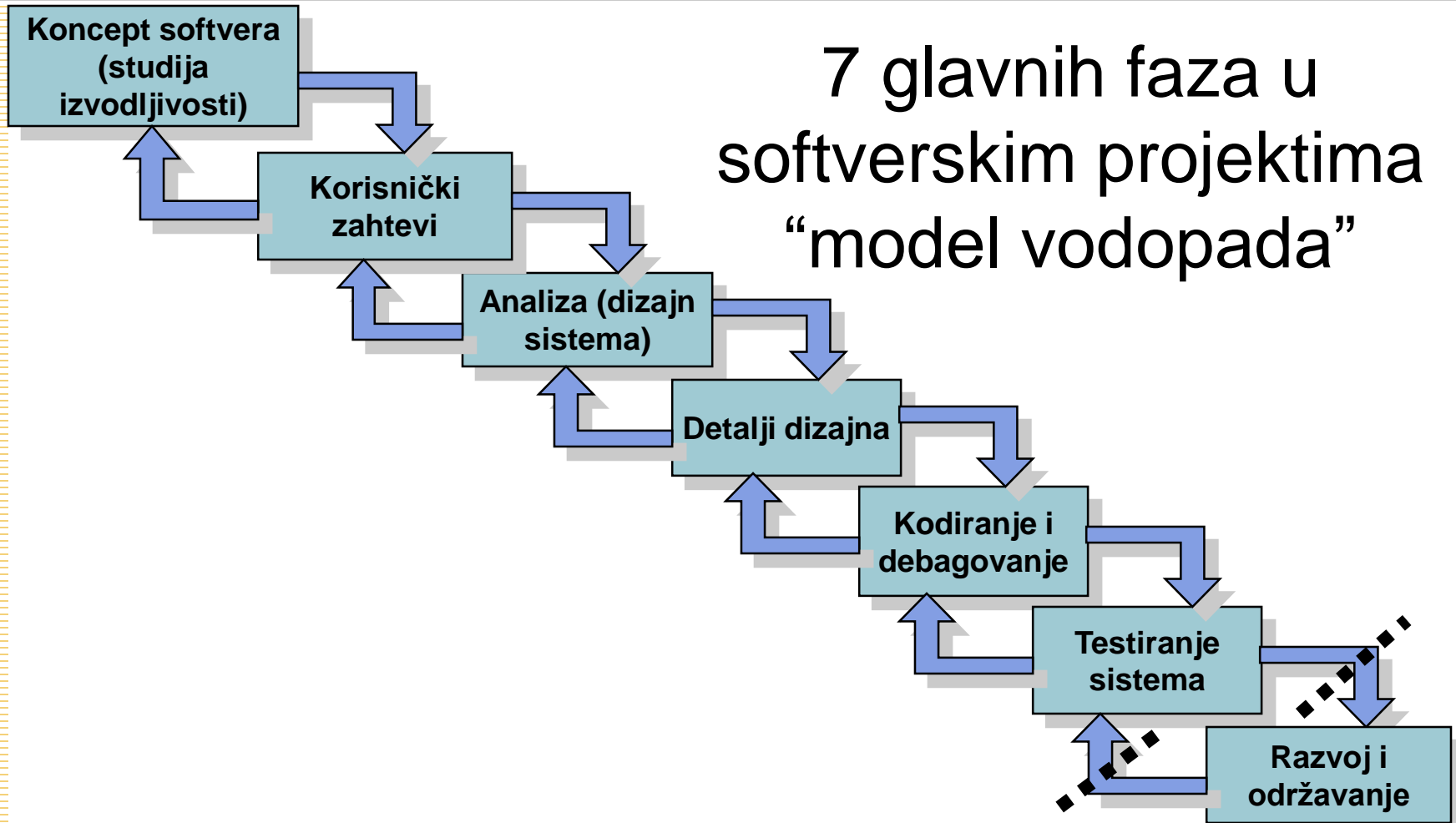


ŽIVOTNI CIKLUS SOFTVERA



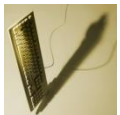
Izvršavanje projekta - standardno

7 glavnih faza u softverskim projektima
“model vodopada”



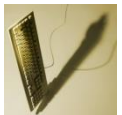
Spiralni model

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



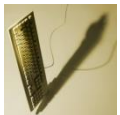
Prototip softvera (1)

- 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



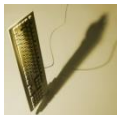
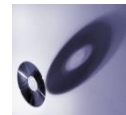
Prototip softvera (2)

- Veliki problem kod prototipa je kontrolisanje promena za sugestije koje dobijamo od strane 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



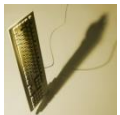
Inkrementalna isporuka

- Sistem treba podeliti na manje komponente, 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?



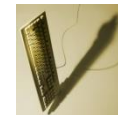
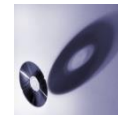
Plan inkrementalne isporuke modula

- Priroda svakog modula i redosled kojim će moduli biti isporučeni korisnicima, treba da budu isplanirani.
- 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.
- Šta treba koristiti:
 - standardni jezik višeg nivoa
 - standardni operativni sistem
 - male module
 - promenljive parametre (podaci da mogu da se zamene bez intervencije programera)
 - standardni sistem za upravljanje bazom podataka

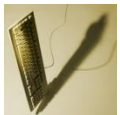
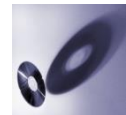


Inkrementalni plan

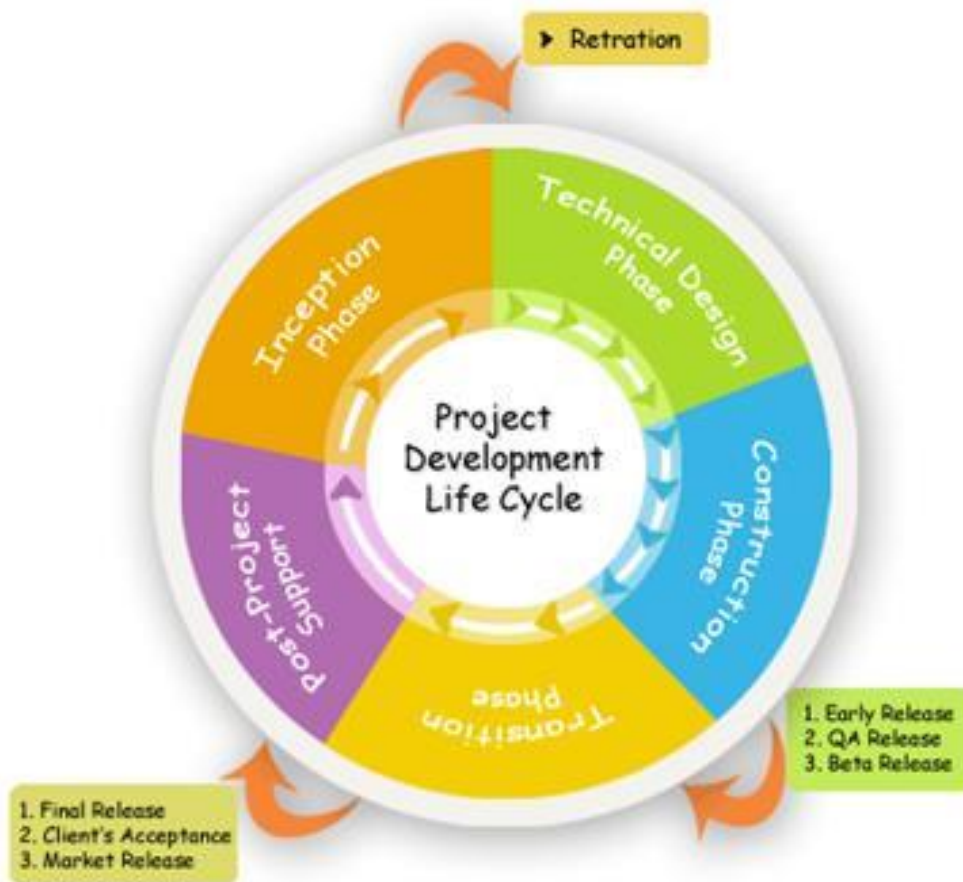
- Primer: novi sistem koji zamenjuje stari sistem i prvi moduli treba da budu delovi starog sistema (koristimo podatke iz stare baze podataka i pravimo neki GUI)
- Odnos Vrednost-Cena se koristi da se odredi redosled razvoja modula:

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

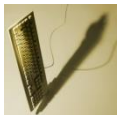
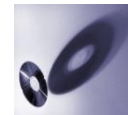
- Na primer V može imati vrednost 1-10 koja prikazuje vrednost za korisnika, a C sa vrednošću 0-10 predstavlja cenu (troškove).
Što je veći ovaj odnos, prioritet je veći.
- Da li cena (troškovi) mogu biti 0?



Izvršavanje projekta - RUP



- Incepcija
- Elaboracija (tehnički dizajn)
- Konstrukcija
- Tranzicija
- Podrška nakon završetka projekta



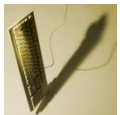
Životni ciklus razvoja softvera ISO12207

- ISO/IEC 12207 (iz 2008) - Systems and software engineering – Software life cycle processes" definiše 43 sistema i softverska procesa
- Šest glavnih procesa:
 - Faza akvizicije (proposal, zahtevi, ugovori,...)
 - Faza postavke (planiranje projekta)
 - Faza razvoja (kreiranje dizajna, kodiranje, izvršavanje testova nad modulima, integracionog i sistemskog testiranja)
 - Operativna faza (instalacija i pomaganje korisnicima u radu)
 - Faza održavanja (poboljšanja, izmene, dopune)
 - Faza destrukcije



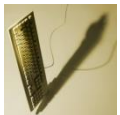
Analiza zahteva

- 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 i zahtev za kvalitetom (koliko dobro funkcionalnosti treba da rade)



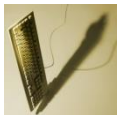
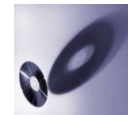
Dizajn arhitekture

- Identifikovati komponente sistema
- Postojeće komponente
- Nove komponente
- Da li komponenta mora biti samo softver?



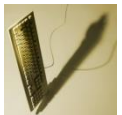
Kodiranje i testiranje

- Detaljniji dizajn = svaku soft.komponentu čini određeni broj softverskih jedinica, koje se mogu odvojeno kodirati i testirati. Dizajn tih jedinica se radi posebno.
- 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



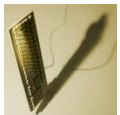
Instalacija i podrška

- Instalacija je proces osposobljavanja novog sistema i puštanje u rad.
- Šta obuhvata instalacija?
- Šta obuhvata podrška?



Extreme programming (XP)

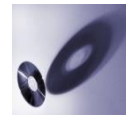
- Ekstremno programiranje je metodologija za razvoj softvera koja je okrenuta poboljšanju kvaliteta softvera i reakciji na promjenjene zahteve kupaca.



Neki praktični saveti - agilne metodologije

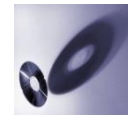
Saveti:

- 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



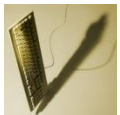
Kategorizacija projekata

- Obavezni vs dobrovoljni korisnici
- Informacioni vs embedded sistemi
- Vrlo su bitni ciljevi projekta!



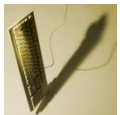
Stakeholders - učesnici projekta

- Učesnici projekta su ljudi koji imaju neki interes na projektu.
- Interni
- Eksterni
- Eksterni van tima i organizacije



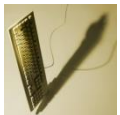
Šta obuhvata menadžment?

- Planiranje - šta će biti urađeno
- Organizovanje - pripremanje za projekat
- Izbor zaposlenih - pravi ljudi na pravim mestima
- Upravljanje - davanje instrukcija
- Monitoring - nadgledanje napretka
- Kontrolisanje - preduzimati akcije da se isprave nedostaci
- Inoviranje - korišćenje novih rešenja
- Prikazivanje - biti u kontaktu sa klijentima, korisnicima, programerima, dobavljačima i drugim stakeholderima



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 projekta 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

