

FER
Zavod za telekomunikacije

Poslijediplomski studij
za stjecanje doktorata
znanosti

Konkurentni sustavi

Konkurentnost u programskim procesima

Ak.g. 2008./2009.

Siječanj 2009.

FER

Sadržaj predavanja

- Konkurentno inženjerstvo
- Konkurentnost u modelima programskih procesa
- Brzi razvoj programskog proizvoda
- Konkurentnost u fazama životnog ciklusa programskog proizvoda
- Podrška razvoju programskog proizvoda
 - Upravljanje konfiguracijom
 - Mjerena programskih proizvoda i procesa
- Alati za podršku konkurentnih aktivnosti u procesu

Konkurentni sustavi Siječanj 2009. 2 od 126

FER

Konkurentnost

Konkurentni sustavi Siječanj 2009. 3 od 126

FER

Uvod

Programski proces

- Razumijevanje
- Komunikacija
- Poboljšanje procesa
- Upravljanje procesom
- Automatizirana podrška izvršavanju procesa

Softverska organizacija

Konkurentni sustavi Siječanj 2009. 4 od 126

FER

Programski proces

- ♦ Strukturirani skup aktivnosti koje zahtijevaju npr. *specifikacija, dizajn, validacija, evolucija* itd.
- ♦ Aktivnosti variraju ovisno o organizaciji i sustavu koji se razvija
- ♦ Proces mora biti eksplicitno modeliran - upravljanje
- ♦ **Značajke procesa**
 - Razumljivost
 - Vidljivost
 - Pouzdanost
 - Kvaliteta

♦ Glavno područje za poboljšanje kvalitete programskog proizvoda jer sadrži aktivnosti tijekom kojih se programski proizvod ili neki njegov dio stvarno kreira!!!

Konkurentni sustavi Siječanj 2009. 5 od 126

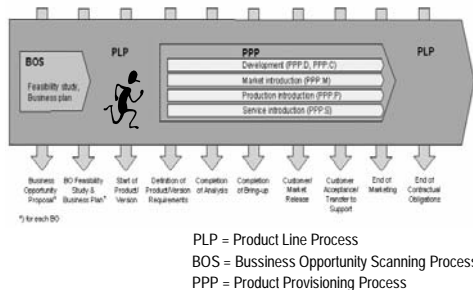
FER

Konkurentne aktivnosti u životnom ciklusu PP

Zahtjevi Dizajn Implementacija Testiranje Isporuca Održavanje

Konkurentni sustavi Siječanj 2009. 6 od 126

Konkurentnost u životnom ciklusu - primjer



Ostali aspekti konkurentnosti



- ◆ Distribuirani razvoj
- ◆ Rad u timu
- ◆ Raznovrsni korisnici
- ◆ **BRZI RAZVOJ**
- ◆ Osobni i timski procesi

Zahtjevi Dizajn Implementacija Testiranje Isporka Održavanje

Konkurentno inženjerstvo



- Istovremeno, interaktivno i interdisciplinarno uključivanje ljudi različitih tehničkih područja (dizajn, proizvodnja, podrška) u zajedničke projekte s ciljem skraćivanja razvojnog ciklusa održavajući pouzdanost, performanse proizvoda, kvalitetu i podršku korisnicima.

B.S. Dhillon. Engineering and Technology Management Tools and Applications. Artech House, 2002.

Konkurentno inženjerstvo



- KI postoji odavno, u svojoj suvremenoj formi početak 1980-tih – Ford Motor Company, Taurus model



Ciljevi konkurentnog inženjerstva



1. Poboljšati kvalitetu proizvoda
2. Smanjiti troškove razvoja proizvoda
3. Smanjiti troškove proizvodnje
4. Smanjiti vrijeme do pojave na tržištu
5. Poboljšanje kompetitivnosti
6. Smanjiti troškove testiranja
7. Povećati granice zarade
8. Smanjiti troškove održavanja

Osnovna načela konkurentnog inženjerstva



- Strukturiranje posla
 - za razliku od računala ljudi ne mogu raditi istovremeno na nekoliko zadataka
 - sistematsko strukturiranje posla ili radne okoline tako da se pojedini zadatak može izvršiti neovisno od strane čovjeka ili računala

Osnovna načela konkurentnog inženjerstva



- Dojam vlasništva
- Postojanost ciljeva i svrhe
- Sklonost timskom radu
- Uzajamno razumijevanje
- Ujednačavanje znanja
 - velika domena: pomagala za podršku donošenja odluka u kombinaciji s bazom ljudskog znanja

Osnovna načela konkurentnog inženjerstva



- Rano otkrivanje problema
- Rano donošenje odluka
- stalna analiza i otkrivanje potencijalnih problema/rizika, pogotovo tijekom početnih faza razvoja

Primjena principa KI



- Čimbenici koji definiraju stupanj i potrebe za KI unutar organizacije:
 - veličina
 - tip proizvoda
 - razvojni i proizvodni procesi
 - fizička lokacija grana kompanije

Timovi unutar konkurentnog inženjerstva



- Suradni timovi
- Članovi koriste svoje ekspertno znanje za postizanje ciljeva tima

Ciljevi:

- pojačana suradnja
- efikasna komunikacija
- česte provjere razvoja
- dodjela odgovornosti

Suradnja u timu



Elementi:

- suradnja
- komunikacija
- kompromisi
- koordinacija
- privrženost
- konsenzus
- kontinuirano poboljšanje

Preduvjeti uspješnog rada u timu



- Preduvjeti vezani uz posao
 - Posao kojeg tim treba obaviti treba biti eksplicitno definiran
 - Posao treba biti razumljiv i smislen
 - Tim treba biti dobro upoznat s poslom koji treba obaviti



Preduvjeti uspješnog rada u timu



- Preduvjeti vezani uz tim
 - Članovi tima trebaju biti jasno definirani
 - Članovi se trebaju međusobno poznavati (ili imati mogućnost upoznati se na početku projekta)
 - Članovi trebaju znati vlastitu ulogu u projektu i uloge ostalih članova
 - Članovi su svjesni svoje odgovornosti i odgovornosti ostalih članova
 - Članovi trebaju imati znanja i vještine da izvrše zadani posao
 - Članovi trebaju biti motivirani



Preduvjeti uspješnog rada u timu (2)



- Preduvjeti vezani uz nadzor
 - Nadzor nad izvršavanjem posla
 - Nadzor nad aktivnostima tima
 - Nadzor nad odnosima u timu (rješavanje problematičnih situacija, međusobno upoznavanje)
- Tim treba imati nadzor nad aktivnostima koje poduzima za izvršenje zadanog posla
- Tim ima kontrolu nad procesom koji koristi
- Tim treba biti obaviješten o napretku posla



Preduvjeti uspješnog rada u timu (3)



- Radni uvjeti tima
 - prostor
 - načini komunikacije
 - tehnologija (obuka!)
 - dostup literaturi i ostalim izvorima informacija
 - nagrada za izvršeni posao
 - ostale pogodnosti



Kohezija tima



- Čvrsta povezanost članova tima u radnu cjelinu
 - tim je smješten na istu fizičku lokaciju i provodi puno vremena zajedno
 - članovi komuniciraju slobodno i često
 - surađuju, uzajamno se poštuju i podupiru jedni druge
 - članovi dijele iste ciljeve i vrijednosti, te imaju slične prioritete
 - u timovima s malom kohezijom: članovi djeluju kao pojedinci, bez velike interakcije s ostalima

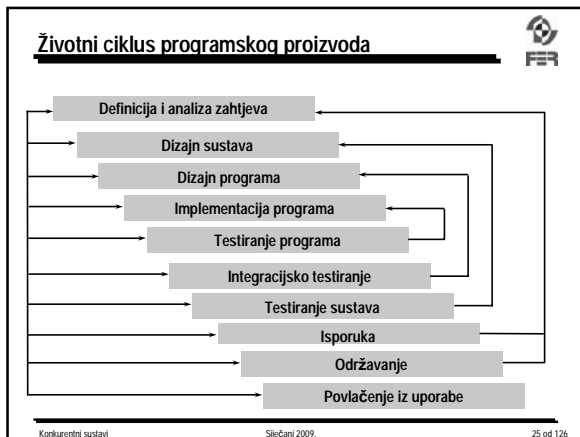
Visoko efikasni tim



- Postoji odanost timu i želja da se ostvare zadani rezultati
- Postoji i izražen je identitet tima
- Članovi tima imaju potrebne kompetencije
- Članovi posjeduju vještine koje se nadopunjuju s vještinama ostalih članova
- Postoji međusobno povjerenje i uzajamna ovisnost
- Komunikacijski kanali unutar tima su poznati i efikasni
- Postoji i užitak u izvršavanju zadanog posla.

Konkurentnost u modelima programskih procesa





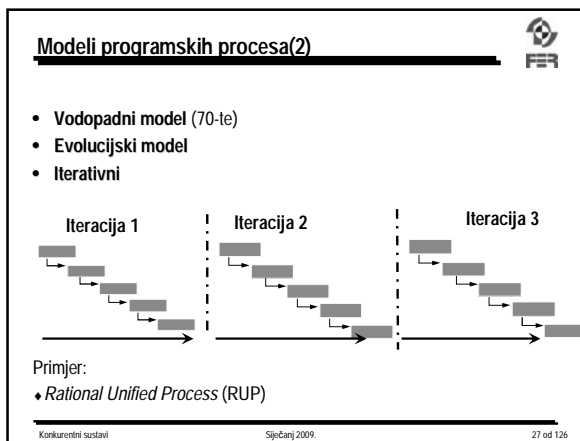
Modeli programskih procesa

- Raznolikost *procesa razvoja programskog sustava*
- Ne postoji univerzalni programski proces!
- Izbor procesa razvoja ovisi o tipu i veličini programskog proizvoda

Primjeri:

- ♦ poslovna aplikacija - aplikacija za obradu informacija u stvarnom vremenu
- ♦ mala aplikacija (1 programer) – veliki programski sustav (multinacionalna kompanija)
- ♦ Različite organizacije - različiti procesi za razvoj istog tipa programskih proizvoda.

Konkurentni sustavi Siječanj 2009. 26 od 126

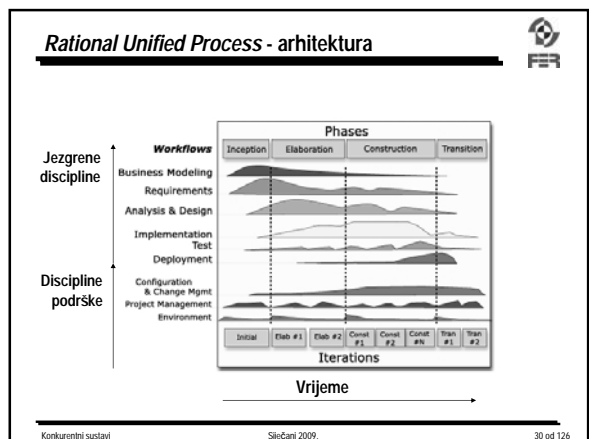
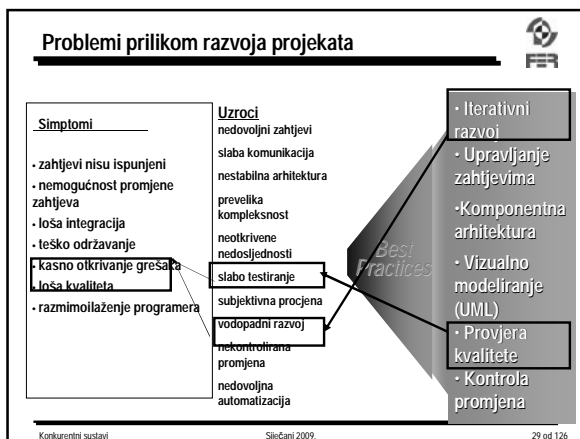


Rational Unified Process

Software Best Practices

Organizirani i dokumentirani skup načela, metoda i procesa koji povećavaju kvalitetu i produktivnost razvoja programskog proizvoda.

Konkurentni sustavi Siječanj 2009. 28 od 126



Uvod



Brzi razvoj (*Rapid Application Development - RAD*)

- kraće vrijeme do isporuke na tržište
- zadržana kvaliteta prog. proizvoda
- veći napori na početku projekta
- Jedan od glavnih principa RAD-a: uvođenje konkurentnih aktivnosti (izbjegavanje aktivnosti, rano izvještavanje o rezultatima)

Konkurentne aktivnosti:

- ♦ postojanje informacija
- ♦ koordinacija
- ♦ kontrola napretka procesa i proizvoda.

Brzi razvoj



Pretpostavke brzog razvoja

- U nekim situacijama 80% rješenje može se proizvesti u 20% vremena potrebnog za izradu cijelog rješenja.
- Poslovni zahtjevi se mogu implementirati u potpunosti čak i ako neki funkcijski zahtjevi nisu implementirani.
- Prihvatljivost se može postići prihvaćanjem min. skupa korisničkih zahtjeva.

Primjena:

- samostalne aplikacije
- sustavi čije performanse nisu kritične
- pouzdanost nije kritična
- tehnologija starija od 1 godine

Brzi razvoj (2)



Za uspjeh:

- potreban dobar balans metoda, pomagala, osoblja i upravljanja

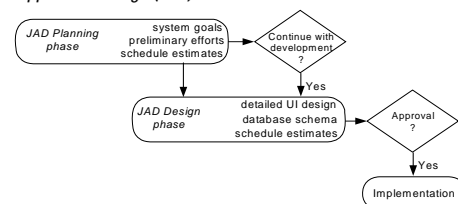
RAD o(ne)mogućavaju:

- tip programskog proizvoda (složenost domene/aplikacije)
- projekt (arhitektura, infrastruktura)
- osoblje (vještine, menadžment)
- platforma

Modeli brzog razvoja



- Vodopadni model s preklapanjem faza
- Vodopadni model s paralelnim podprojektima
- *Daily Build and Smoke Test*
- *Joint Application Design (JAD)*:



Konkurentnost u fazama životnog ciklusa programskog proizvoda



Konkurentne aktivnosti prikupljanja zahtjeva



Praćenje zahtjeva



Requirements traceability

Praćenje s obzirom na izvor zahtjeva:

- Veza između zahtjeva i osoba/dokumenta koji ih specificiraju

Praćenje s obzirom na načelo zahtjeva:

- Veza zahtjeva i objašnjenja zašto je zahtjev specificiran

Praćenje s obzirom na druge zahtjeve:

- Veza između svih zahtjeva koji su međusobno ovisni, veza je dvosmjerna

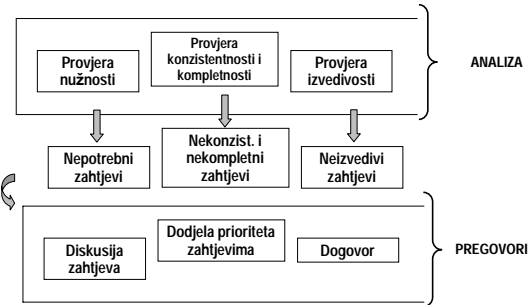
Praćenje s obzirom na arhitekturu:

- Veza zahtjeva i podsustava u kojima su zahtjevi implementirani, posebno važno kod distribuiranog razvoja.

Praćenje s obzirom na dizajn:

- Veza zahtjeva i specifične sklopovske ili programske komponente koja se primjenjuje u implementaciji zahtjeva.

Konkurentne aktivnosti analize zahtjeva



Upravljanje zahtjevima



ID	Ime	Tip	Status	Planirani početak	Stvarni početak	Stabilnost	Teškoća
1	Mogućnost stvaranja jedinstvenog identifikatora	Funkcionalni zahtjev	1	2009-01-01	2009-01-01	1	1
2	Obzirom na zahtjev	nefunkcionalni zahtjev	2	2009-02-01	2009-02-01	4	4
3	Obzirom na zahtjev	Funkcionalni zahtjev	3	2009-03-01	2009-03-01	2	2
4	Obzirom na zahtjev	zastarjela zahtjev	4	2009-04-01	2009-04-01	3	3
5	Mogućnost stvaranja jedinstvenog identifikatora	Funkcionalni zahtjev	1	2009-05-01	2009-05-01	4	4
6	Mogućnost stvaranja jedinstvenog identifikatora	Funkcionalni zahtjev	3	2009-06-01	2009-06-01	5	5
7	Obzirom na zahtjev	nefunkcionalni zahtjev	2	2009-07-01	2009-07-01	3	3
8	Obzirom na zahtjev	Funkcionalni zahtjev	3	2009-08-01	2009-08-01	3	3

Primjer pomagala za zahtjeve



- Rational RequisitePro, podrška prikupljanju i upravljanju
- mogućnost praćenja - matrica, stablo

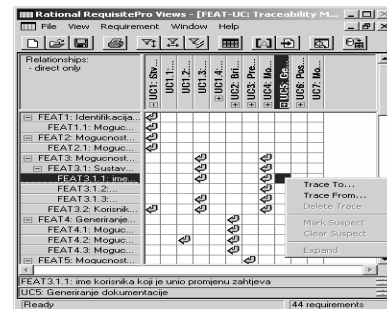


Primjer pomagala za zahtjeve (2)



Requirements	Priority	Status	Planned Start	Actual Start	Stability	Difficulty
FEAT1: Identifikacija zahtjeva	High	Approved	1	1	Medium	Medium
FEAT1.1: Mogućnost stvaranja jedinstvenog identifikatora	High	Approved	1	1	Medium	Medium
FEAT2: Mogućnost stvaranja jedinstvenog identifikatora	High	Approved	1	1	Medium	Medium
FEAT2.1: Mogućnost stvaranja jedinstvenog identifikatora	Medium	Approved	1	1	Medium	Medium
FEAT3: Mogućnost stvaranja jedinstvenog identifikatora	Medium	Approved	1	1	Medium	Medium
FEAT3.1: ime korisnika koji je unio promjenu zahtjeva	Medium	Approved	1	1	Medium	Medium
FEAT3.1.1: ime korisnika koji je unio promjenu zahtjeva	Medium	Approved	1	1	Medium	Medium
FEAT3.1.2: datum i vrijeme kada je promjena zahtjeva	Medium	Approved	1	1	Medium	Medium
FEAT3.1.3: naloga promjena zahtjeva	Medium	Approved	1	1	Medium	Medium
FEAT3.2: Korisnik može susopisati veze među zahtjevima	Medium	Approved	1	1	Medium	Medium
FEAT4: Generiranje dokumentacije	Medium	Approved	1	1	Medium	Medium
FEAT4.1: Mogućnost generiranja specifične zahtjeva u obliku	Medium	Approved	1	1	Medium	Medium
FEAT4.2: Mogućnost generiranja specifične zahtjeva u obliku	Low	Proposed	3	1	Medium	Medium
FEAT4.3: Mogućnost generiranja specifične zahtjeva u obliku	Medium	Approved	1	1	Medium	Medium
FEAT5: Mogućnost pretraživanja zahtjeva	High	Approved	1	1	Medium	Medium
FEAT5.1: Mogućnost pretraživanja zahtjeva po prioritetu	High	Approved	1	1	Medium	Medium
FEAT5.2: Mogućnost pretraživanja zahtjeva po statusu	High	Approved	1	1	Medium	Medium
FEAT5.3: Mogućnost opisa like svih podzastarjelih zahtjeva	High	Approved	1	1	Medium	Medium
FEAT5.4: Opisi funkcioniranja	High	Approved	1	1	Medium	Medium
FEAT6: Mogućnost biranja zahtjeva	High	Approved	1	1	Medium	Medium
FEAT6.2: Mogućnost pregleda podzastarjelih zahtjeva	High	Approved	1	1	Medium	Medium
FEAT6.3: Mogućnost biranja zahtjeva	High	Approved	1	1	Medium	Medium

Primjer pomagala za zahtjeve (3)



Distribuirani razvoj



Distribuirani razvoj (1)



Pokretači:

- Trend globalizacije
- Dinamika tržišta (kratki proizvodni ciklusi)

Ostale mogućnosti:

- Lokalizacija
- *Pool* raspoloživih vještina

Tipovi distribuiranih projekata:

- *Open Source* (dobrovoljni razvijatelji dodaju promjene, jezgri tim razvijatelja prihvaća ili odbija promjene)
- Virtualne korporacije
- Organizacije s više sjedišta

Distribuirani razvoj (2)



- Distribuirani timovi
- ☺ Ušteda troškova i vremena
- ☺ Brzi dostup do eksperata
- ☺ Brzi dostup do sklopovlja
- Pomagala:
 - video-konferencije
 - pomagala grupne podrške

Distribuirani razvoj (3)



- Ayoma:
"Distribuirani konkurentni razvoj zamjenjuje sekvencijalni centralizirani razvoj u velikim telekomunikacijskim programskim projektima."

Prednosti:

- Drastično smanjenje vremena razvoja
- Veća produktivnost

Distribuirani razvoj (4)



Osnovni izazovi distribuiranog razvoja:

- Komunikacija
- Koordinacija
- Vremenski pomak
- Različite kulture
- Različite interpretacije
- Poznavanje ljudi
- Zajednički repozitorij

Distribuirani razvoj (5)



Problemi:

- ☹ Suradnja
 - (podrška neformalnoj komunikacija i pregovorima - sustavi grupne podrške)
 - povećavaju se troškovi i smanjuju mogućnosti u distr. razvoju
- ☹ Upravljanje dokumentacijom
 - Upravljanje konfiguracijom (paralelni rad na više verzija)
 - Nekonzistentnosti: ne mogu se izbjeći, trebaju se identificirati i pratiti
 - Onemogućen centralizirani pristup problemima! (☹?)

Distribuirani razvoj (6)



Upravljanje projektom

- Modeliranje i definiranje procesa
- Planiranje i utvrđivanje rokova
- Praćenje procesa

Upravljanje znanjem

- Eksplicitno znanje
- "Pronalaženje eksperata"

Lokalizacija

- Različiti centri distribuiranog razvoja imaju različite potrebe
- Različiti centri imaju primjenjuju različite metode/pomagala

PP s javno dostupnim kodom – "Open source software"



- Primjer distribuiranog razvoja
- *Open source SW* pruža korisnicima :
 - slobodu korištenja
 - slobodu mijenjanja ili prilagodavanja
 - slobodu distribucije.
- Uz program se distribuira:
 - izvorni kod
 - licenca – pravni dokument koji osigurava ove slobode
- Drugi naziv je slobodan softver (*free software*)

Nastanak "Open Source" softvera



- 1983. godine Richard Stallman pokreće *Free Software Foundation* - "*Free as in free speech, not as in free beer*"
- 1991. godine Linus Torvalds počinje pisati jezgru operacijskog sustava kojeg naziva Linux
- 1998. godine nastaje *Open Source Initiative*
- Razvojem Interneta dolazi i do punog korištenja *Open Source Software*-a.

Primjeri "Open Source" softvera



- *Linux*
 - operacijski sustav temeljen na Unixu
- *Apache*
 - *Web* poslužitelj korišten na preko 50% poslužitelja
- *Sendmail*
 - poslužitelj koji procesira oko 80% ukupne elektroničke pošte
- *OpenOffice*
 - skup uredskih alata

Razvoj "Open Source" softvera



- Tipičan *open source* projekt ima ove osobine:
 - razvoj je baziran na Internetu
 - vodeći tim razvijatelja (do 20ak članova) donosi odluke konsenzusom
 - svi korisnici mogu se uključiti u proces razvoja i dati svoj doprinos
 - projekt se u svakom trenutku može razgranati na dva projekta.

Prednosti razvoja "Open Source" softvera



- ☺ Omogućava neovisne recenzije:
 - moguć velik broj recenzenata
 - posljedica je kvalitetniji i pouzdaniji programski proizvod
- ☺ Omogućava lakše otkrivanje i brže ispravljanje neispravnosti
 - velik broj razvijatelja može paralelno tražiti i ispravljati neispravnosti
 - posljedica je stabilniji i pouzdaniji programski proizvod.

Problemi razvoja "Open Source" softvera



- ⊗ Broj razvijatelja može biti velik i oni su najčešće prostorno dislocirani - komunikacija.
- ⊗ Može se dogoditi da više razvijatelja istovremeno radi na istom dijelu koda.
- ⊗ Različiti projekti mogu dijeliti dio koda

Sustavi grupne podrške - općenito



- Sustavi koji osiguravaju programsku podršku članovima nekog tima za postizanje zajedničkog cilja.
- Članovi tima:
 - rade na različitim mjestima i u različito vrijeme
 - ista ili neka druga radna organizacija
 - surađuju
 - komuniciraju
 - stalni - privremeni članovi tima
 - zahtjeva se brzi odaziv
 - nemogućnost zajedničkog sastanka
 - različiti izvori znanja.

Sustavi grupne podrške (3)



Prednosti sustava grupne podrške:

- ⊗ Paralelna obrada informacija
- ⊗ Sudjelovanje velikog broja različitih sudionika
- ⊗ Brze diskusije
- ⊗ Agresivni - povučeni sudionici - u jednakom statusu
- ⊗ Uštede na putovanjima.

Nedostaci sustava grupne podrške:

- ⊗ gubitak konteksta i fokusa diskusije
- ⊗ mogućnost nerazumijevanja/hedorečenosti.

Sustavi grupne podrške (2)



Anytime/Anyplace Collaboration

- omogućeno preko Web-a

Groupware:

- programska podrška suradnji u timskom distribuiranom radu
- sustavi elektroničkih sastanaka
- sustavi elektroničkih konferencija

Okvir vrijeme/prostor

- Isto vrijeme - isti prostor: *Decision Room*
- Isto vrijeme - različiti prostor: Video konferencija
- Ostale kombinacije vrijeme - prostor: Internet

Sustavi grupne podrške (3)

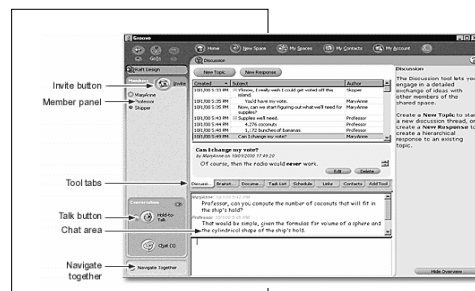


Dvorana za donošenje odluka (Decision room)

- 12-30 umreženih osobnih računala
- računalo-poslužitelj (*facilitator computer*)
- veliki projekcijski sustav
- potreba za medijatorom sastanka



Primjer - Groove sustav



Distribuirane inspekcije PP



Distribuirane inspekcije PP



- Ispitivanje programskog proizvoda
- Inspekcija - statički pregled koda
- Tijek odvijanja inspekcije:
 - Inicijalizacija inspekcije
 - Priprema za inspekciju
 - Odvijanje inspekcije
 - Analiza rezultata
- Provjera specifikacije, dizajna i koda

Sustavi grupne podrške za inspekcije PP



- Rezultati (+):
 - ⊙ 60 do 80% pronađenih neispravnosti
 - ⊙ Ušteda do 50% u parametru osoba-sat
 - ⊙ Probijanje rokova 90% rjeđe
 - ⊙ Organizacije troše 10 do 20% resursa na inspekcije programskog proizvoda
- Rezultati (-):
 - ⊙ Ne znamo rezultate nakon inspekcije

Sustavi grupne podrške za inspekcije PP



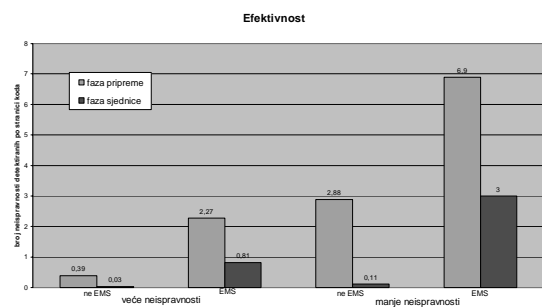
- Vrste neispravnosti kod programskog proizvoda:
 - Veće neispravnosti (*majors*)
 - Manje neispravnosti (*minors*)
- Parametri kod inspekcija podržanih sustavima grupne podrške:
 - Efektivnost (*effectiveness*)
 - Učinkovitost (*efficiency*)
 - Učinak (*yield*)

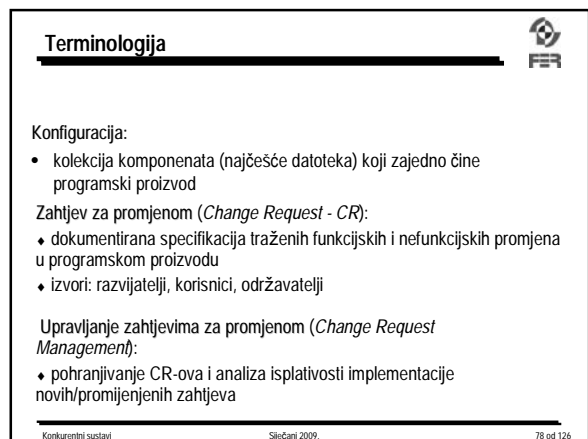
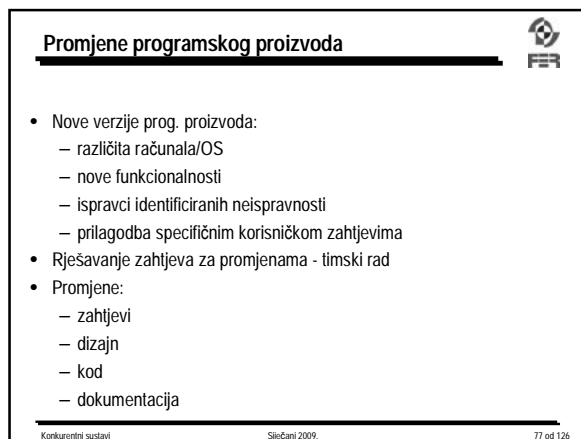
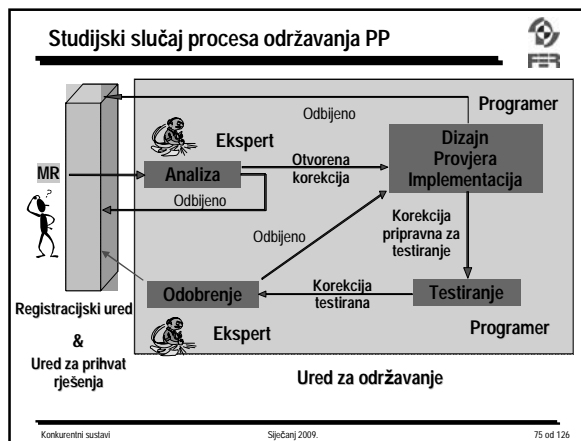
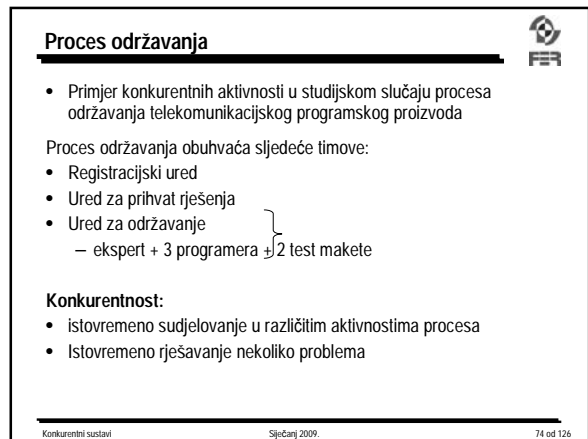
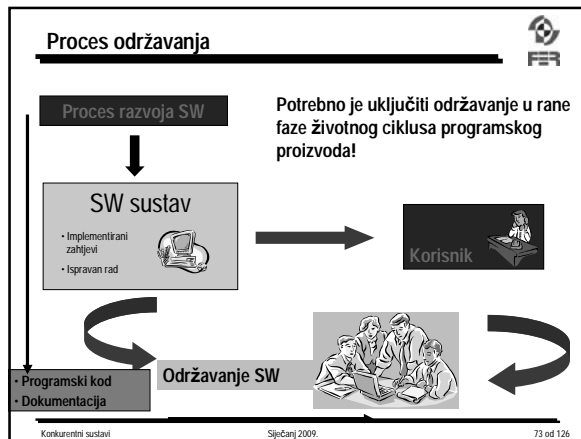
Sustav za podršku elektroničkih sastanaka



- Tip sustava grupne podrške u kojim se komunikacije između sudionika uspostavljaju elektronički
- Lista neispravnosti u čvrsto definiranom formatu (kategorizacija)
- "Tihe sobe"
- Sudionici inspekcije ne moraju biti na istom dijelu koda

Rezultati istraživanja inspekcija podržanih sustavima za podršku elektroničkih sastanaka





Terminologija (2)



Verzija

- Instanca programskog proizvoda koja se funkcijski razlikuje od drugih instanci koje se odnose na isti zahtjev programskog proizvoda.

Isporučka (*Release*)

- Instanca programskog proizvoda koja se distribuira korisnicima izvan razvojnog tima
- Izvršni kod, konfiguracijske datoteke, instalacijski programi, dokumentacija (elektronička, papirnata)
- CD, WWW

Terminologija (3)



- Početni sustav (*Baseline*) - referentan za sve eventualne nadogradnje

- Ako revizije kreira više od jednog korisnika:
obostrani isključivi pristup

Check-out: zaključana datoteka za ostale korisnike

Check-in: pohranjivanje nove revizije

Upravljanje programskom konfiguracijom



- *Software Configuration Management (SCM)*
- Upravljanje promjenama i složenosti programskog proizvoda
 - Komunikacija
 - Rad u timu
 - Distribuirani razvoj
- Razvoj i primjena standarda i procedura za upravljanje evoluirajući prog. proizvodom
- Dio procesa upravljanja kvalitetom

Definicija SCM



- *Upravljanje konfiguracijom programskog proizvoda je disciplina upravljanja i kontroliranja promjena tijekom evolucije programskog proizvoda.* (IEEE Std. 1042-1987)
- Pomaže bolju koordinaciju pojedinaca koji rade u istoj fazi razvoja nekog dijela programskog proizvoda
- Osigurava informacije vezane uz:
 - trenutno stanje verzija
 - ispravak neispravnosti
- Omogućava oporavak u slučaju gubitka bilo kojeg dijela prog. proizvoda.

Problemi vezani uz SCM



Simultano ažuriranje

- Dva ili više razvijatelja rade istovremeno na istom kodu

Podijeljeni kod

- Unosi se promjena u kod o kojem ovisi rad drugih razvijatelja (treba ih pravovremeno obavijestiti ili im dostaviti promijenjeni kod).

Zajednički kod

- promjene se unose u biblioteke komponenata za višestruko korištenje

Kontrola promjena



- koristi se za upravljanje repozitorijom komponenti
- omogućava pohranjivanje različitih komponenti prog. proizvoda i njihovih verzija sigurno i jednoznačno

- Nakon svake promjene kreira se *revizija* neke datoteke

dat.1, dat.2, dat.3, ...

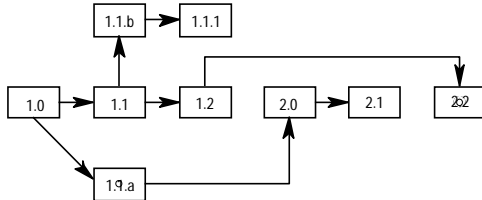
History - bilježi se tko je napravio reviziju i kada, eventualno dodavanje posebnih komentara (zašto?) - zasebna datoteka

- Revizije ne moraju nužno biti sekvencijalne
- Nova grananja: *dat.2, dat.2.1, dat.2.2*

Struktura generiranja novih verzija



Primjer:



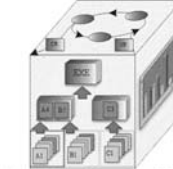
Praktični aspekti kvalitete procesa



Configuration management

- Upravljanje zahtjevima za promjenom (*Change Request Management*): pohranjivanje CR-ova i analiza isplativosti implementacije zahtjeva

Upravljanje zahtjevima za promjenom (CRM)



Mjerenja

Upravljanje konfiguracijom (CM)

- Upravljanje konfiguracijom (*Configuration Management*): podjela poslova, kontrola verzija, upravljanje resursima, paralelni razvoj, koordiniranje aktivnosti

Upravljanje zahtjevima za promjenom



ZAHTJEV ZA PROMJENOM

Formulari zahtjeva za promjenom

DOKUMENTIRANJE

PRIHVAT PROMJENE?

Planiranje SCM-a



- rane faze projekta
- za velike sustave:
 - veliki broj procesne dokumentacije (zahtjevi, specifikacije, kod, test-podaci, korisnički priručnici. itd.)
- Odrediti nivo formalizma (ovisi o veličini projekta, prog. proizvoda, timu, SW organizaciji)
- Poseban naglasak - dokumentacija za održavanje

SCM plan



Definiranje:

- tipovi dokumenata
- standardi imenovanja
- odgovornosti za CM
- politika kontrole promjena i upravljanje verzijama
- pomagala
- održavanje CM zapisa

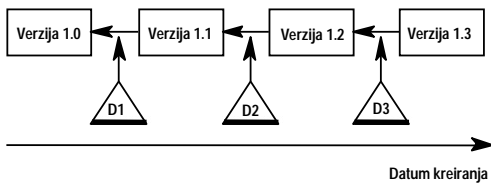
Pomagala za SCM



RCS

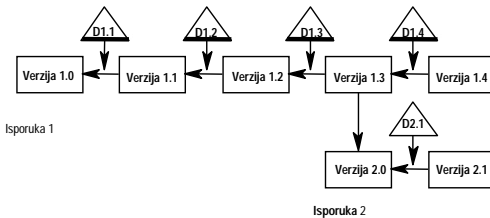
- staro pomagalo, upravljanje različitim verzijama sustava
- CVS
- kasnije će biti detaljno objašnjeno
- ClearCase
- više servera, modeliranje procesa, mehanizmi provjere

RCS - Revision Control System



- Poshranjuju se samo razlike između verzija i početnog programskog proizvoda (*baseline*)

Paralelni razvoj s RCS-om



Isporuka 1

Isporuka 2

Sustav konkurentnih verzija (CVS - Concurrent Versions System)

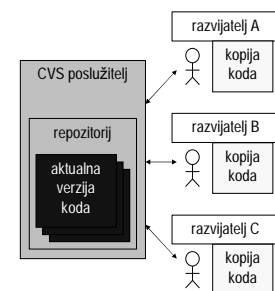


- CVS je sustav za praćenje verzija izvornog koda
- omogućuje:
 - dohvat starih verzija koda – npr. zato da bi se vidjelo koja je promjena u kodu uzrokovala neispravnost
 - da više razvijatelja istovremeno radi na istom kodu
 - grananje na dva ili više projekta i po potrebi njihovo kasnije "stapanje" u jedan projekt
- također omogućuje praćenje verzija i kod:
 - tekstualnih dokumenata
 - HTML stranica

Princip rada CVS-a



- originalne verzije koda se čuvaju u CVS repozitoriju
- razvijatelji rade na kopijama koje se poslije "stapaju" u novu verziju originala
- "stapanje" i otkrivanje mogućih konflikata je automatizirano



Dobre i loše strane CVS-a



- dobre strane:
 - olakšava traženje neispravnosti
 - eliminira potrebu za strogom koordinacijom razvijatelja
 - automatizira spajanje koda različitih razvijatelja
 - CVS je *open source* softver
- loše strane:
 - ne snalazi se najbolje s binarnim datotekama
 - nema ugrađen sustav za praćenje neispravnosti

Konkurentnost u upravljanju projektima



Projekt - definicija



Projekt je privremeni usmjereni napor poduzet kako bi se razvio jedinstveni proizvod, usluga ili ostvario neki drugi specifični rezultat.
[PMBOK 2004]

Svojstva projekta



- Svojstvo *privremenosti*:
 - vrijeme trajanja (početak, kraj)
 - projektni tim
 - tržište i sl.
- Kraj projekta:
 - ciljevi ostvareni
 - znanje o nemogućnosti ostvarivanja ciljeva projekta
 - ne postoji više potreba za projektom.
- Svojstvo *jedinstvenosti*
 - prisutnost ponavljajućih elemenata u projektima ne mijenja njegovu jedinstvenost (na primjer korištenje prethodnog iskustva).

Svojstva projekta



- Svojstvo *privremenosti*:
 - vrijeme trajanja (početak, kraj)
 - projektni tim
 - tržište i sl.
- Kraj projekta:
 - ciljevi ostvareni
 - znanje o nemogućnosti ostvarivanja ciljeva projekta
 - ne postoji više potreba za projektom.
- Svojstvo *jedinstvenosti*
 - prisutnost ponavljajućih elemenata u projektima ne mijenja njegovu jedinstvenost (na primjer korištenje prethodnog iskustva).

Svojstva projekta (2)



- Svojstvo *progressivne elaboracije*:
 - ujedinjuje svojstva privremenosti i jedinstvenosti
 - razvoj projekta u koracima
 - opseg projekta se definira na početku
 - tijekom trajanja projekta povećava se znanje projektnog tima o cilju i očekivanim rezultatima
 - opseg projekta se redefinira
 - projekt postaje eksplicitniji
 - potreban nadzor opsega

Rezultati projekta



- Jedinstveni proizvod, usluga ili ostvarenje nekog drugog rezultata
- Materijalni proizvod koji se može kvantificirati - konačni proizvod ili pojedina komponenta
- Mogućnost pružanja neke usluge (poslovne usluge koje podržavaju razvoj, distribuciju i sl.)
- Dokumentacija
- Znanje (istraživački projekti) - potrebno ga je adekvatno opisati.

Upravljanje projektom - definicija



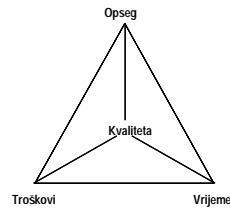
Upravljanje projektom je primjena znanja, tehnika, vještina, i pomagala na definirane aktivnosti projekta kako bi se ostvarili ciljevi projekta.
[PMBOK 2004]

- Projektni menadžer: osoba **odgovorna** za realizaciju projektnih ciljeva

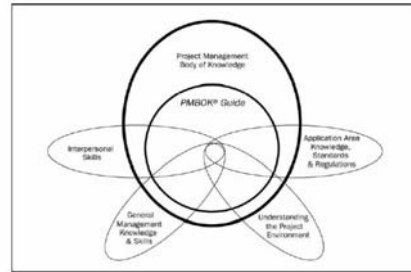
Upravljanje projektom - aktivnosti



- Identifikacija:
 - zahtjeva
 - jasnih i realnih ciljeva
- Balansiranje zahtjeva za kvalitetom, opsegom, vremenom i troškovima



Ekspertna područja tima za upravljanje projektom



Područja ekspertize projektnog tima [PMBOK 2004]

Projektna okolina



Projekti imaju pozitivan /negativan utjecaj na okoline:

- Kulturalna/sociološka okolina - način na koji projekt utječe na ljude i kako ljudi utječu na projekt
- Međunarodna/politička okolina - zakoni, običaji, politika, nacionalni praznici, putovanja
- Fizikalna okolina - okoliš, geografija

Potrebno je razmotriti projekt u kontekstu njegove okoline, s obzirom na svrhu projekta.

Podprojekti



- Podjela projekata na manje cjelina kako bi se olakšalo upravljanje.
- Podprojekti se mogu smatrati zasebnim projektima

Primjer:

- Projektni proces - pojedina faza životnog ciklusa projekta
- Zahtjevi za specifičnim vještinama pojedinih ljudskih resursa
- Tehnologije (npr. automatsko testiranje programa)

Najčešći mitovi vezani uz upravljanje projektima (1)



"To je samo mali projekt. Upravljanje nije potrebno. Treba se odmah samo baciti na posao."

- Svaki projekt treba upravljanje.
- U malim projektima menadžer može imati i druge uloge
- Za svaki projekt, pa i najmanji, treba postojati plan
- Nadzor projekta: ciljevi, rokovi

Najčešći mitovi vezani uz upravljanje projektima (2)



"Upravljanje projektom će odnijeti znatan dio vremena zbog dokumentacije."

- Djelomično točno.
- Razina formalizma u projektu:
 - tip projekta
 - metoda/sustav za upravljanje
- Vještina projektnog menadžera da prilagodi potrebe projekta i razinu formalizma specifičnim okolnostima projekta.

Najčešći mitovi vezani uz upravljanje projektima (3)



"Svaki projekt kojim se dobro upravlja ima vjerojatnost uspjeha 100%."

- Nije moguće nadzirati sve aspekte projekta.
- Upravljanje ne može kompenzirati nedostatak znanja/vještina tima, lošu/pogrešnu tehnologiju, nedostatak povratne veze unutar tima/od strane korisnika i sl, nerealne rokove i neadekvatnu podršku sponzora.

Najčešći mitovi vezani uz upravljanje projektima (4)



Glavna zadaća projektnog menadžera je osigurati zadovoljenje rokova.

- To je jedna u nizu zadaća i odgovornosti.
- Osiguranje i nadzor kvalitete, ostanak u granicama budžeta, izgradnja tima, ...
- Primjena pomagala za upravljanje projektima ne može nadomjestiti znanje, vještine, intuiciju, praksu projektnog menadžera, ali ih može nadopuniti.

Definicija projektnog procesa



Projektni proces:

- Skup međusobno povezanih aktivnosti koje se izvršavaju kako bi se ostvario prethodno specificirani skup rezultata, proizvoda ili usluga.
- **Procesi zajednički većini projekata veći dio vremena** (pokretanje, planiranje, izvršavanje, praćenje i nadzor, završavanje projekta)
- **Procesi orijentirani prema proizvodu** (specifični za određenu domenu, životni ciklus proizvoda)

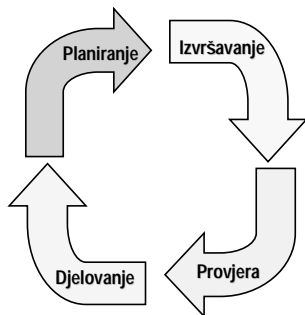
Odabir projektnih procesa



- **Projektni menadžer, u suradnji s projektnim timom, odgovoran je za određivanje koji procesi su prikladni za određeni projekt, te potrebne razine formalizma i strogosti u implementaciji tih procesa.**

- Projekt može biti vođen na različite načine:
 - složenost, veličina, trajanje, iskustvo tima, pristup resursima, zrelost organizacije u PM-u, industrija, domena

Načelo identifikacije projektnih procesa

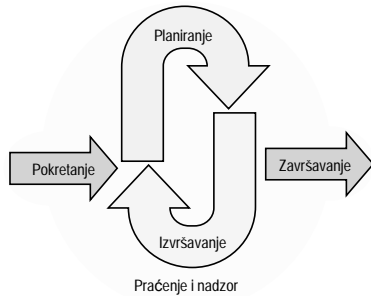


Grupe procesa zajedničke većini projekata

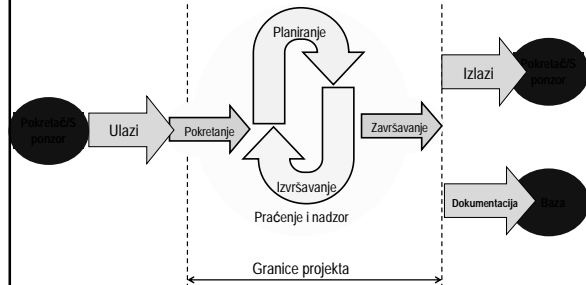


- Grupa procesa za pokretanje projekta
- Grupa procesa za planiranje
- Grupa procesa za izvršavanje
- Grupa procesa za praćenje i nadzor
- Grupa procesa za završavanje projekta

Interakcije između projektnih procesa



Granice projekta



Definicija životnog ciklusa projekta



Životni ciklus projekta:

- Skup uglavnom slijednih faza projekta koje se definiraju s obzirom na potrebe nadzora projekta.
- Faze koje povezuju početak i kraj projekta.
- Može biti dokumentiran određenom projektnom metodologijom.

Životni ciklus projekta



Određuje:

- Posao koji treba napraviti u određenoj fazi
- Rezultat svake pojedine faze
- Način provjere rezultata (inspekcije)
- Sudionike pojedine faze
- Način nadzora i odobranja faze

Rezultati izvršenja projektne faze



- zaključenje pojedine faze ne znači automatsko započinjanje nove
- svaka nova faza se formalno inicira specifikacijom što je u njoj dozvoljeno i očekivano - nadzor, upravljanje

Inspekcije po završetku pojedine faze:

- suglasnost za zaključenje dotične faze
- inicijacija sljedeće
- *phase exit, phase gate, kill point*

Definiranje životnog ciklusa projekta



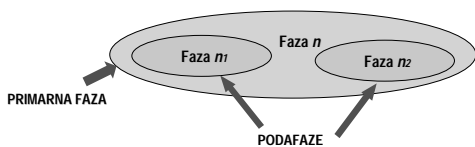
- Ne postoji jedinstvena definicija
- Mogućnosti:
 - standardizacija za sve projekte unutar pojedine kompanije
 - određivanje za svaki projekt posebno
 - preferirani životni ciklusi ovisno o industrijskoj okolini projekta.

Faze životnog ciklusa projekta

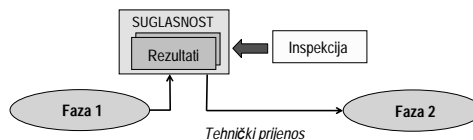


Rezultat izvršavanja pojedine faze:

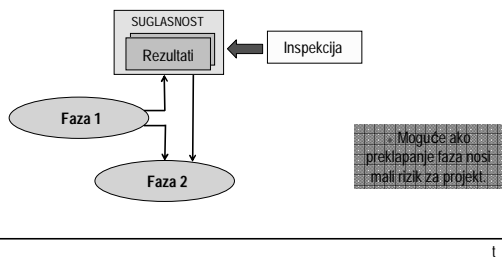
- mjerljiva količina posla koja se može provjeriti
- specifikacija, studija izvodljivosti, izvještaj, dokumentacija, prototip
- za nadzor odvijanja projekta i postizanje njegovog konačnog cilja



Odnos faza životnog ciklusa projekta



Preklapanje faza životnog ciklusa projekta



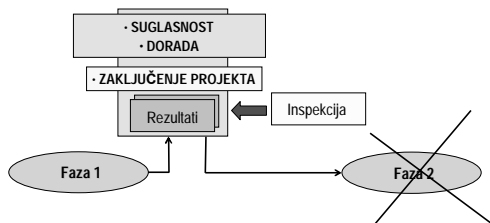
Faze životnog ciklusa projekta



Podjela na faze ovisi o:

- veličini projekta
 - složenosti
 - rizicima
 - toku novca
- Svaka podfaza treba imati definirane svoje rezultate - povezani su s rezultatima primarne faze

Rezultati izvršenja projektne faze



- Formalno zaključivanje pojedine faze ne znači automatsko započinjanje nove

Slijed faza u projektu



		Tim za upravljanje projektom			
Ulazi	Ideja				
Faze	Inicijalna	Konstrukcijske		Završna	
Izlazi važni za upravljanje projektom	Prijedlog Izvjeshaj o opsegu	Plan Osnova za usporedbu	Progres	Suglasnost Prihvaćenost	Isporuca
Projektni rezultati	PROIZVOD USLUGA REZULTAT				