**Sustav** - uređeni poredak međuzavisnih komponenti povezanih za postizanje zajedničke svrhe ili cilja:

* **Fizički sustav** materijalne naravi
* **Apstraktni sustav** konceptualni, proizvod ljudske mašte

**Informacijski sustav** - skup međusobno povezanih komponenti koji prikuplja, obrađuje, pohranjuje i pruža informacije potrebne za obavljanje poslovne zadaće

**Slojevi poslovnih inf. sus.:**

* **Transaction Processing System (TPS)** evidencija i obrada podataka o poslovnim transakcijama
* **Management Information System (MIS)** upravljački obrada transakcijskih podataka, Decision Support System (DSS) za potporu odlučivanju
* **Decision support system (DSS)** odlučivanje na temelju podataka iz različitih izvora
* **Executive Information System (EIS)** koristi se za donošenje strateških odluka

**Softver** - programska potpora, oprema (programi + podaci)

**Aplikacija** - primjenjena programska potpora, namjenski program

**Programsko inženjerstvo** - sistematičan, discipliniran i mjerljiv pristup razvoju, primjeni i održavanju softvera

**Informacijski sustav** - sustav aplikacija za upravljanje ljudskim aktivnostima

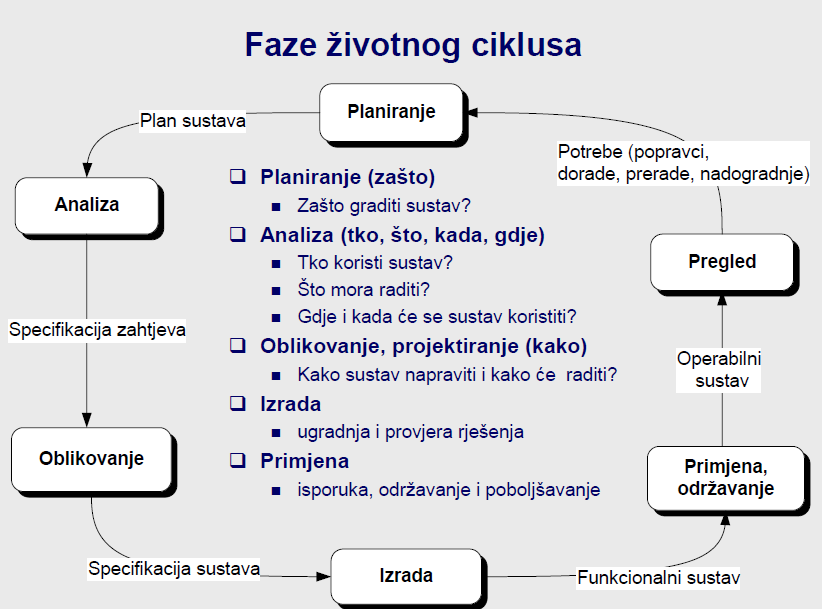
**Informacijsko inženjerstvo** - inženjerski pristup izradnji IS i upravljanju informacijama

**Projektiranje IS** - slijed izrade fizičkog i logičkog modela postojećeg i budućeg IS

**Sistemska analiza** - proučavanje poslovanja s ciljem preporuke poboljšanja i specificiranja zahtjeva na rješenje

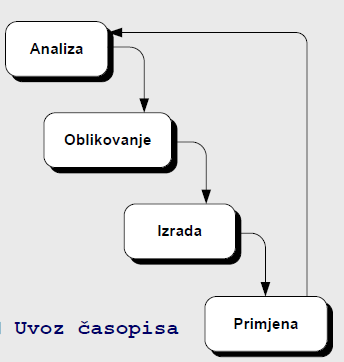
**Sistemski dizajn** - projektiranje sustava

**SA + SD = informacijsko inženjerstvo**

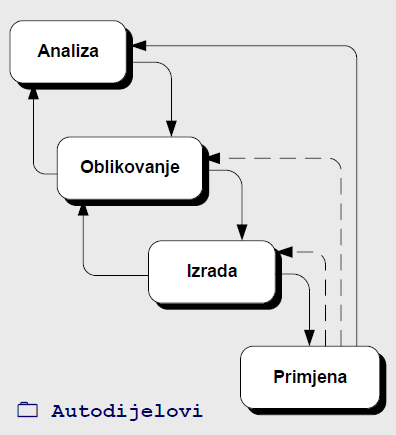


**Modeli razvoja**

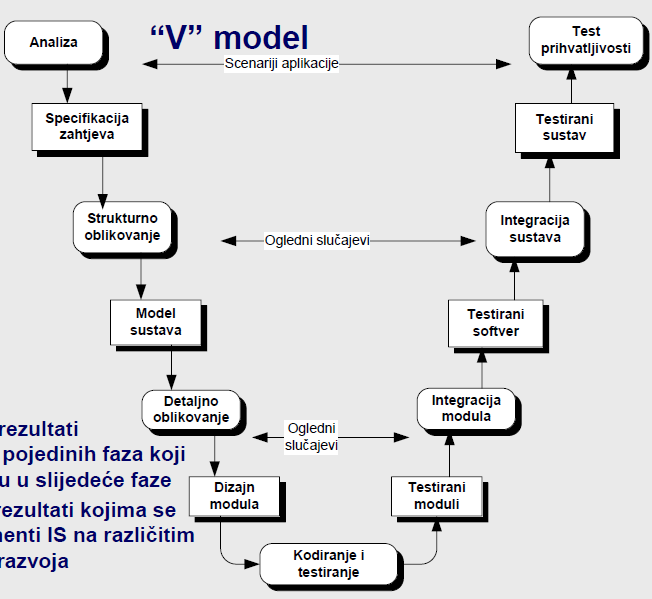
* **Vodopadni klasični** slijedno napredovanje iz faze u fazu, kvaliteta važnija od troškova i rokova
  + + dobro definirano i stabilno okruženje, uhodane ručne obrade ili računarski sustav koji treba unaprijediti, veliki projekti
  + - uvođenje prema gore (moduli, podsustav, sustav), sustav neupotrebljiv dok nije potpuno završen, korisnici prekasno uoče nedostatke



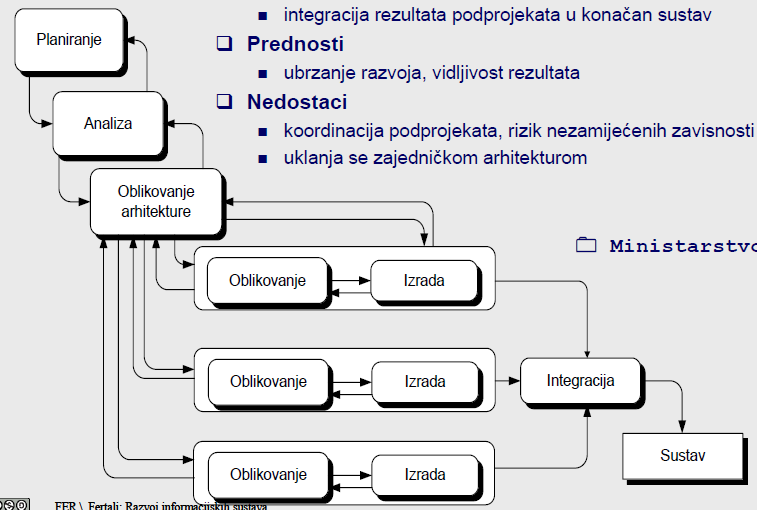
* **Pseudostrukturirani i radikalni vodopadni model** povratna veza i mogućnost promjene prethodnih rezultata, strukturirano programiranje, istovremenost različitih faza



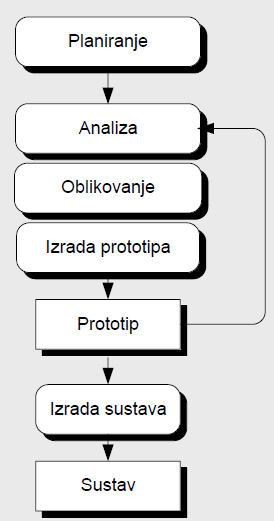
* **V Model** definiraju se rezultati pojedinih faza koji se prosljeđuju u slijedeće faze, određuju se rezultati kojima se testiraju elementi IS na različitm stupnjevima razvoja



* **Paralelni razvoj po podprojektima** integracija rezultata podprojekata u konačan sustav
  + + ubrzanje razvoja, vidljivost rezultata
  + - koordinacija podprojekata, rizik nezamijećenih zavisnosti



* **Brzo prototipiranje**
  + + pokretljivost, prilagodljivost, bolje određivanje zahtjeva, vidljivost rezultata
  + - zaboravljanje da prototip nije pravi sustav, vječni razvoj, izostanak specifikacije pa druge komunikacije



* **Fazni razvoj** slijed verzija koje se isporučuju i generiraju nove zahtjeve
  + + isporuka dijelova prije konačne verzije, ranije vidljiva korist sustava, korisnici brže identificiraju dodatne zahtjeve
  + - korisnici upotrebljavaju sustav koji je namjerno nedovršen, potreba za pažljivim planiranjem
* **Spiralni model** iterativno se analizira rizik i planira naredna iteracija od 6 koraka
  + + praćenje napretka, sigurnost
  + - složenost, trošak

**Organizacijska izvedivost**

* **PIECES koncept** - performance, information, economics, control, efficiency, service

**Postupak procjenjivanja**

* **procjena veličine projekt (metrika projekta)**
  + broj funkcijskih točaka (FP) ili programskih redaka (LOC) procjena broja ulaza, izlaza i datoteka i njihove složenosti npr. 30 LOC/FP za C# -> za 6000FP zahtjeva izradu 180000 LOC, produktivnost za poslovni sustav 500 LOC/čm -> ukupni napor 360čm
  + broj objektnih točaka, procjena broja zaslonskih maski, izvješća i modula za pristup podacima, 500OT uz produktivnost 25 OT/mj. zahtijeva 20čm
* **procjena potrebnog napora u čovjek-mjesecima (čm)**
  + iz procjene veličine projekta i produktivnosti
* **procjena trajanja projekta (vremenski plan)**
  + iz napora i veličine ekipe u kalendarskim godinama ili mjesecima 36čm/6 članova = 6 mjeseci

**optimalno trajanje u mjesecima =**

**faktor 3.0 može varirati od 4.0 do 2.5 ovisno želi li se brže ili sporije odvijati**

npr. za napor 65čm optimalno trajanje 3.0\*651/3 = 12 mjeseci, ako se podijeli s 12 mjeseci to je 5-6 članova tima

**vrijednost novog kupca** -> vrijednost 300 novih kupaca godišnje koji prosječno potroše $500 po proizvodu koji nakon troškova donosi 12% dobiti, godišnja dobit iznosi 300 \* $500 \* 12% = $18000

**vrijednost postojećih kupaca** -> gubitak 100 kupaca od kojih svaki troši $2500 godišnje uz dobit 12%, a za čije reklamiranje treba uložiti $50000 gubitak iznosi = 100\*$2500 \* .12 + $50000 = $80000

**smanjenje cijene rada / ušteda smanjenjem posla** -> smanjimo trajanje zadatka s 5min na 30sec, a radi osoba za $50 na sat, ušteda = (5-30/60)/60 sati po zadatku \* $50 na sat = $3.75 po zadatku

**Sadašnja vrijednost (PV)** - današnja vrijednost onoga što će postati $1 nakon n godina u budućnosti ako se uzmu u obzir kamate I ->

**Neto sadašnja vrijednost (NPV) = (ukupna korist - ukupni troškovi)**

**Postotak povrata investicije (ROI) =**

**Vrednovanje rješenja build vs. buy**

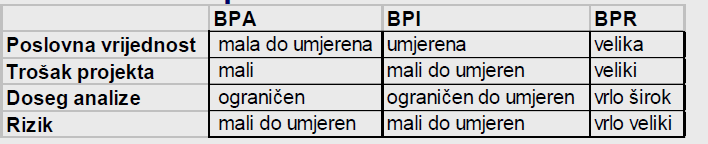
**Ponderirano vrednovanje**

**Automatizacija poslovnih procesa**

* **Business process automation (BPA)** - povećanje učinkovitosti primjenom računalne tehnologije
  + analiza problema - otkrivanje problema sustava i predlaganje rješenja uz pomoć korisnika
  + analiza uzroka - usmjerenost na probleme, a ne na rješenja
  + 2700 putnih naloga godišnje, nalog za putovanje odobravaju voditelj, predstojnik i dekan -> gomilanje naloga u pisarnici, česte gužve na blagajni -> rješenje kreiranje i ažuriranje naloga na mjestu nastanka, automatizacija postupka odobravanja, isplata predujmova putem tekućeg računa
* **Business process improvement (BPI)** - manje promjene radi boljeg iskorištenja tehnologije i učinkovitosti/djelotvornosti
  + analiza trajanja
  + korištenje poslovnih procesa
  + baždarenje i usporedba
  + obračun nastave na poslijediplomskom studiju, jednom godišnje izrađuje se autorski ugovor s općim uvjetima, promjerak potpisanog ugovora dostavlja se u stu slu, naknadno angažirani izvođrači potpisuju dodatak ugovoru, stu slu po obavljenom poslu evidentira obavljeno te generira nalog za isplatu, poboljšanje obrade putnih naloga, izvješće s putovanja temeljem predloška putnog naloga s funkcijom za urudžbiranje – urudžbiranje se obavlja web servisom koji uz atribute pohranjuje i dokument (BLOB)

**Preustroj, preoblikovanje poslovnih procesa**

* **Business process reengineering (BPR)**
* **Business process redesign (BPR)** - radikalni redizajn analizom posljedica, procjenom alternativnih tehnologija, ukidanjem ili zamjenom aktivnosti, analizom troškova i analizom rizika



**Vrste zahtjeva**

* **Poslovni (zašto)** - ciljevi organizacije, poslovni problemi ili potrebe, sve što opisuje financijski ili drugi poslovni probitak za korisnike ili organizaciju
  + sustav mora onemogućiti korištenje subvencije od strane osobe koje nemaju na to pravo, zaradu ilegalnih posrednika, naplatu usluga koje nisu pružene
* **korisnički (zahtjevi krajnjih korisnika)** - opisuju zadatke koje korisnik mora moći obaviti
  + ukinuti plaćanje unaprijed, ukloniti nepotrebne postupke za ostvarivanje prava, smanjiti rizik gubitka ostvarenih prava
* **funkcionalni (što, neko ponašanje)** - očekivana softverska funkcionalnost potpore korisničkih zadataka, korisnik mora moći, sustav treba
  + ne obavlja se storno uplate, radi se storno ulaznih dokumenata, unosom odgovarajućeg dokumenta
* **nefunkcionalni (kako ili kako dobro)**
  + viševalutni rad, svaki zapis sadrži identifikator korisnika i datum unosa i promjene, omogućiti izvoz izvješća u excel formatu

**Zahtjevi na kvalitetu**

* **Dostupnost** -> omjer MTTF/(MTTF+MTTR) srednje vrijeme do kvara, srednje vrijeme do oporavka do kvara
* **učinkovitost** -> stupanj iskorištenja resursa
* **prilagodljivost** -> lakoća dodavanja novih mogućnosti
* **integritet, interoperabilnost, pouzdanost, robustnost, upotrebljivost, lakoća održavanja, prenosivost, ponovna upotrebljivost, podložnost testiranju**

**Problemi pri određivanju zahtjeva**

* **Kuhinjski sudoper** - korisnik navodi brojne forme, pristup uzrokovan pomanjkanjem iskustva korisnika koji ne zna razlučiti bitno
* **Taktika dimne zavjese** - korisnik traži više nego što treba
* **Taktika treba mi isto ali bolje** - nedostaje korisniku volja i znanje, a ponekad oboje
* **korisnik je sklon prešutjeti izuzetke koji su bitni za uspješnu realizaciju**

**Postavljanje prioriteta**

* **Nužno imati (must)**
* **Poželjno imati (should)**
* **Neobavezno svojstvo (could)**
* **Nepotrebno svojstvo (won't)**

**Označavanje zahtjeva**

* **Uzastopni brojevi** - z01, z02, fz01..
* **Hijerarhijsko numeriranje (X,Y,Z)**

**Upravljanje zahtjevima**

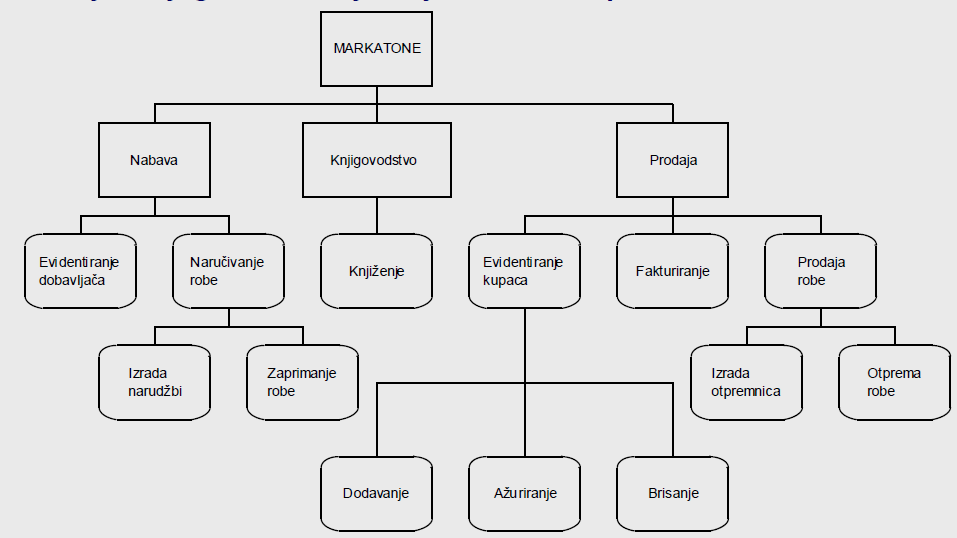
* **Definiranje postupka za promjenu zahtjeva, Uspostava odbora za promjene, Analiza utjecaja promjene zahtjeva, Praćenje promjena na svim proizvodima, uspostava vremenske osnovice i kontrole verzija, praćenje povijesti promjena zahtjeva, praćenje statusa zahtjeva**

**Poslovno pravilo** - instrukcije i logika procedure obavljanja procesa

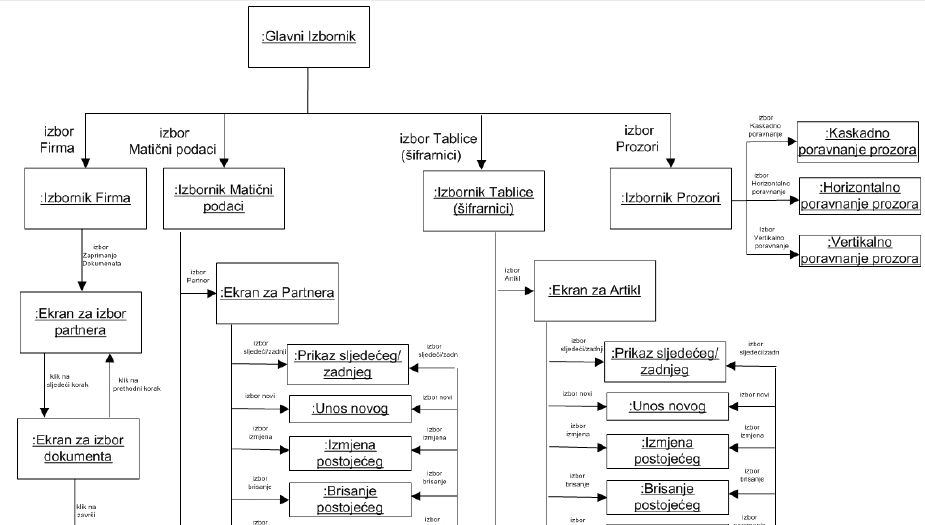
**Poslovna politika** - skup poslovnih pravila

**Funkcionalna dekompozicija**

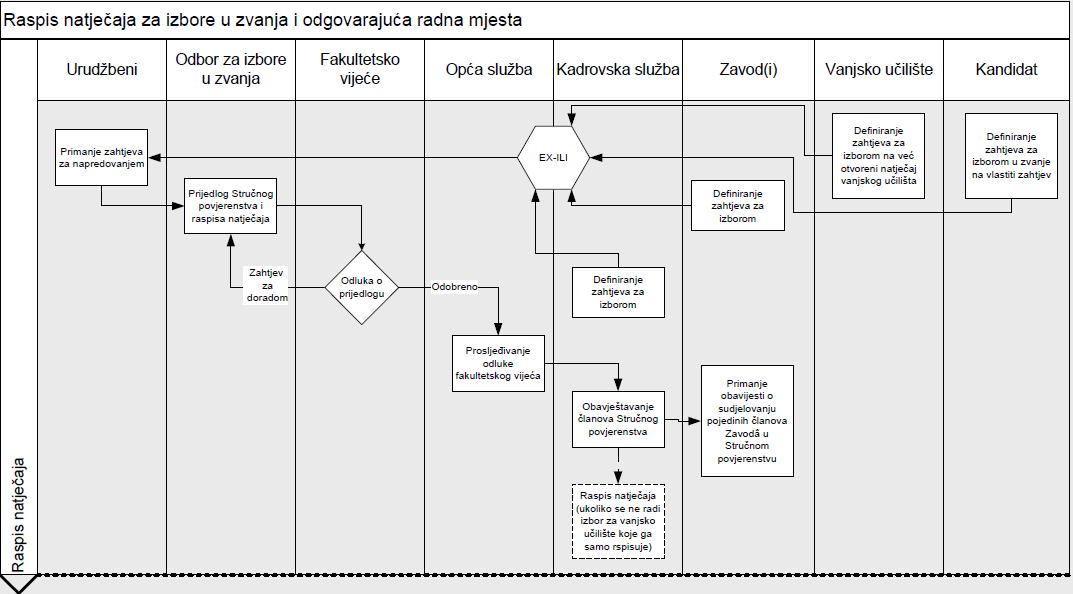
* dijagram dekompozicije funkcija (FDD), hijerarhijski prikaz
* funkcije - označavaju se imenicom (Prodaja, proizvodnja..)
* procesi - glagolski izraz oblika infinitiv+objekt
* **izostaviti procese koji premještaju ili preusmjeravaju podatke bez mijenjanja, pažnju usmjeriti na procese koji nešto računaju, potpomažu odluke, filtriraju ili agregiraju podatke, pokreću druge procese, rukuju podacima**



**Dijagram organizacije** - prikaz strukture organizacije hijerarhijom pravokutnika, svaki je određena uloga ili odgovornost u organizaciji

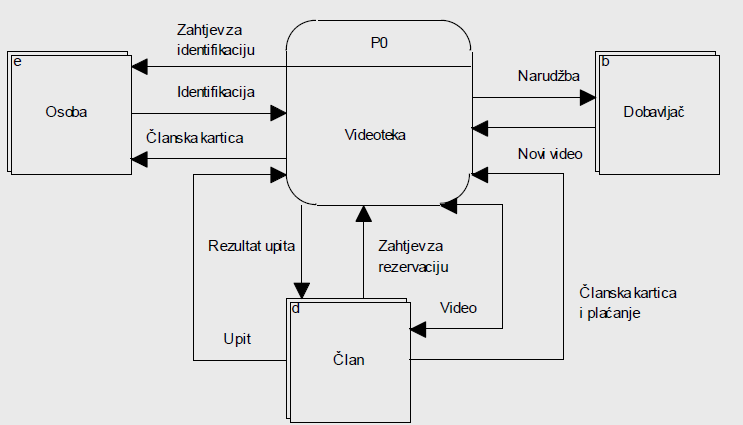
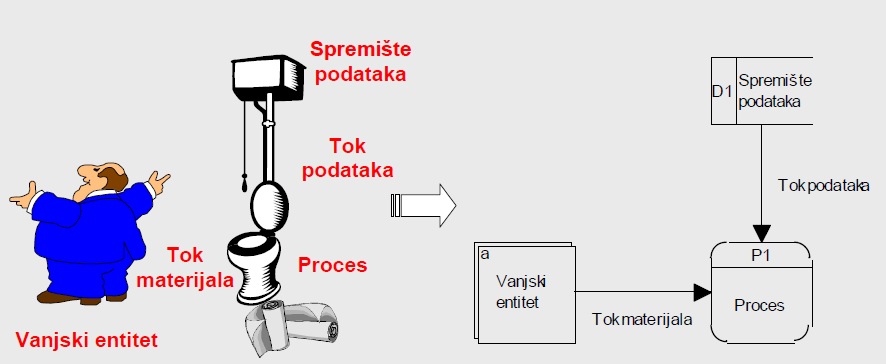


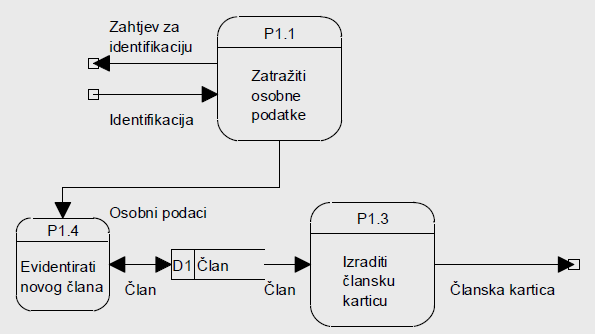
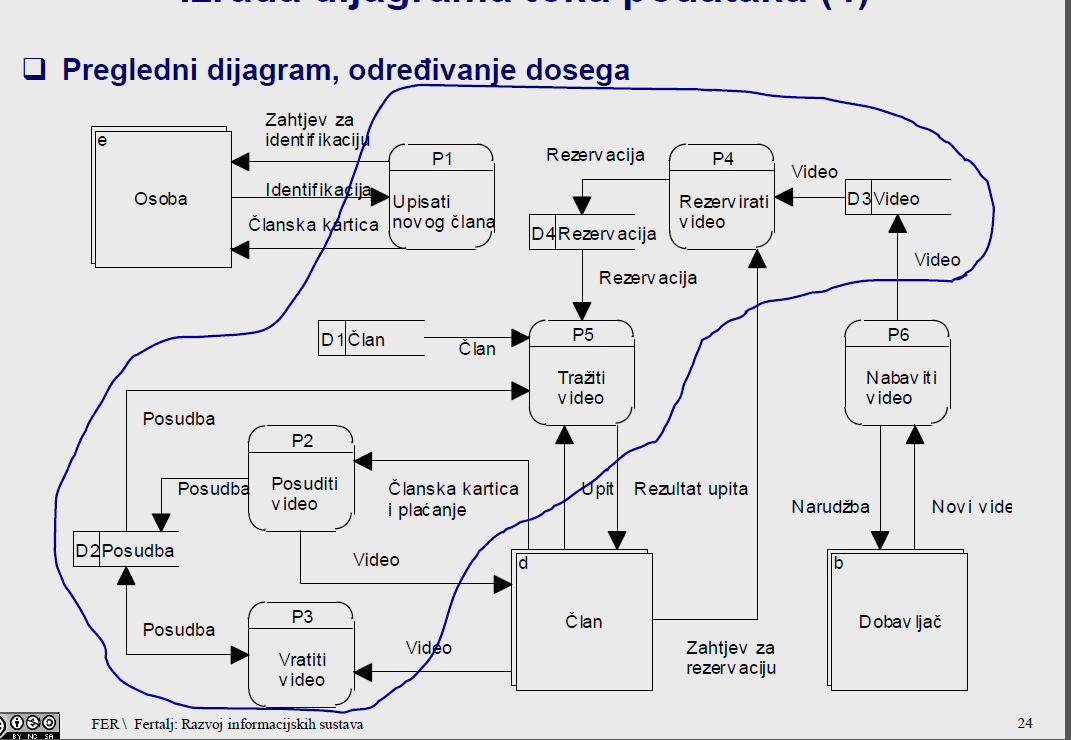
**Modeliranje toka rada (workflow)** - slijed koraka koji obrađuje jednu poslovnu transakciju, definira logiku obrade i precizira nositelja



**Modeliranje toka podataka (DFD)**

* **Tok podataka** - skupovi podataka koji kolaju sustavom, naziv pridjev+imenica
* **Spremište podataka** - organizirani trajni skup podataka, CRUD, naziv imenica
* **Proces** - aktivnost pretvorbe ulaznog u izlazni tok podataka, naziv infinitiv+objekt (prijaviti ispit, prijava ispita)
* **Vanjski entitet** - objekt vanjskog svijeta povezan s promatranim sustavom, određuje granice sustava, naziv imenica (student, kupac, dobavljač)





Razrada osnovnog procesa "Upis novog člana"

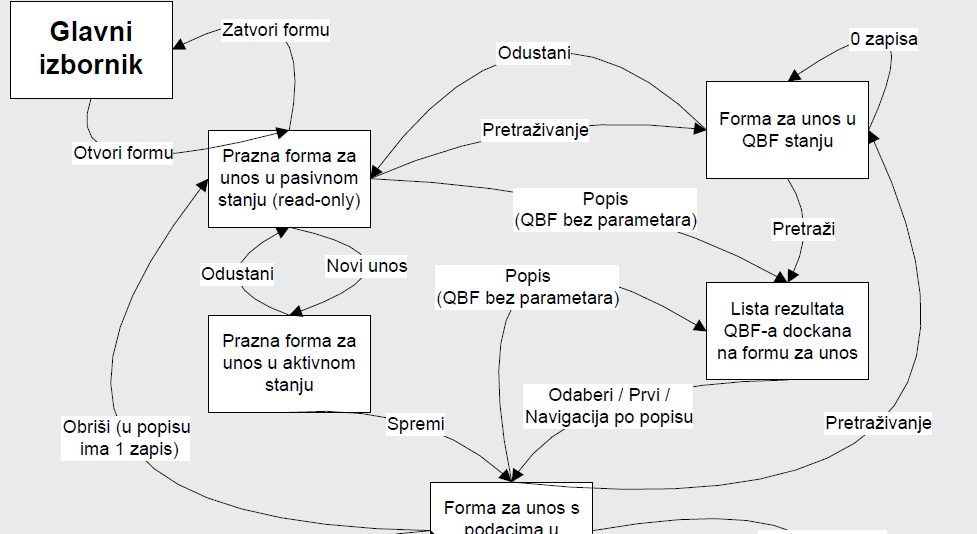
**Modeliranje događaja**

* **Događaj** -zbivanje u sustavu koji vodi ili pokreće procese sustava, sam događaj nije proces nego okidač koji se njime pokreće
* **Vanjski događaji** - potaknuti od strane vanjskih entiteta, zahtjevaju informaciju ili ažuriranje podataka

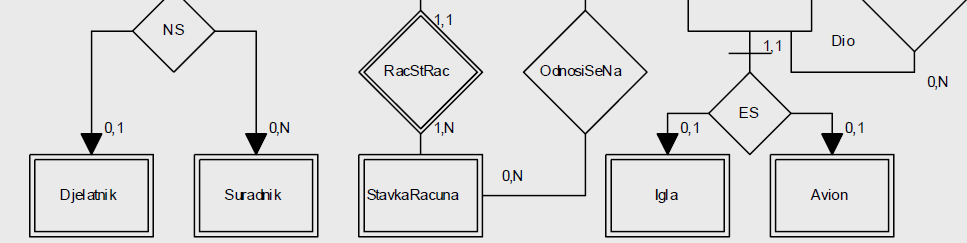
**Dijagram prijelaza stanja**

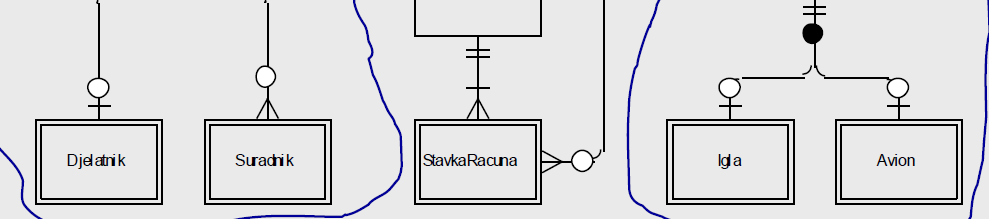
* **stanje** - kumulativni rezultata ponašanja nekog objekta
* **prijelaz** - promjena stanja uzrokovana nekim događajem
* **događaj** - uvjet promjene stanja i akcija koja se obavlja

**Mape dijaloga**



**Entiteti veze**



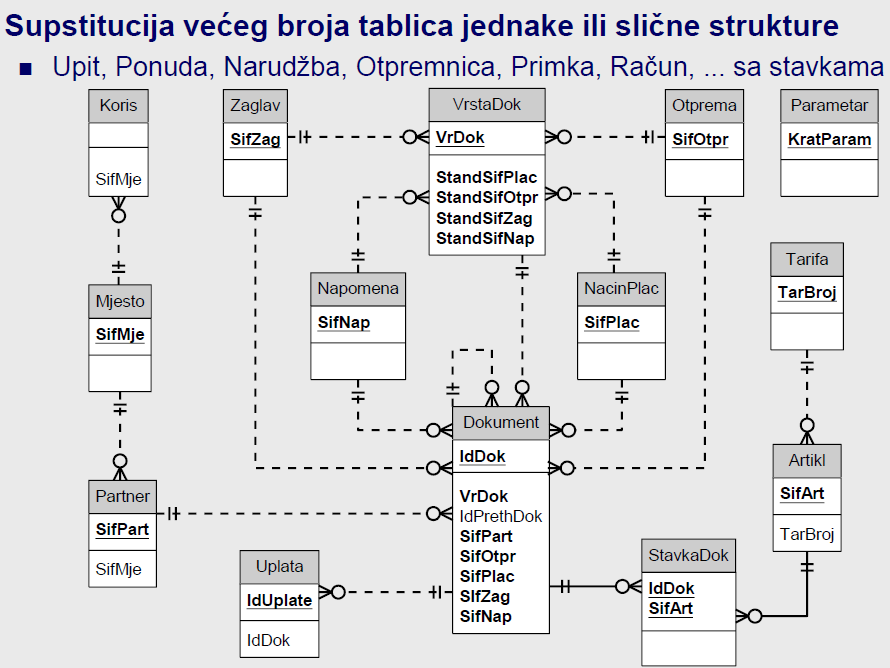


**Šifrarski sustav**

* **Serijske šifre** - brojevi koji se slijedno pridjeljuju svakoj novoj instanci entiteta
* **Blok šifre** - serijski brojevi grupirani prema značenju - 100-199 PAY PER VIEW, 200-299 CABLE..
* **Alfanumeričke oznake** - skup znakova kombiniran s brojevima - HR, DE, IT..
* **Samogovoreće šifre** - svaka znamenka ili grupa znamenki opisuje neko svojstvo instance - JMBG
* **Hijerarhijski kodovi** - podjela u grupe, podgrupe.. itd.
* **Šifre s kontrolnom znamenkom** - zadnja znamenka se izračunava iz svih ostalih - JMBG, Tekući račun..

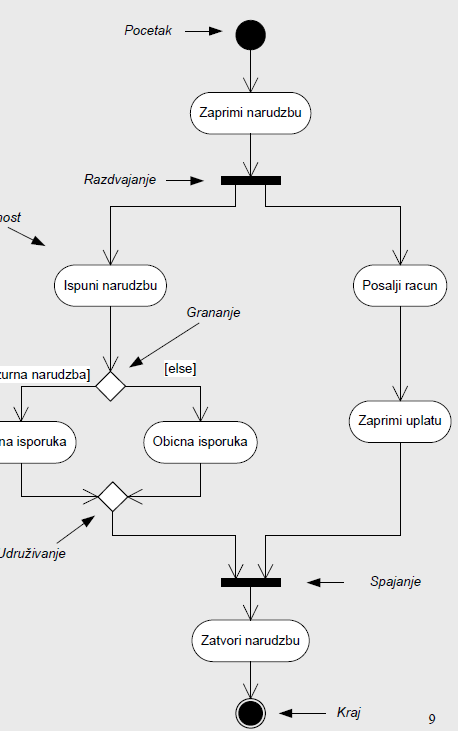
**Meta modeliranje - model za opisivanje drugih modela**

**Metabaza** - DB za pohranu modela

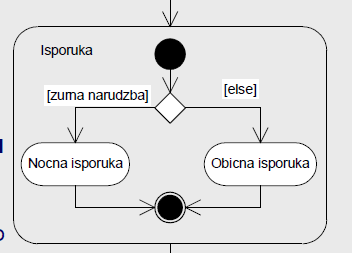


**Dijagram aktivnosti**

* prikazuje ili tok rada između objekata ili tok upravljanja za neku operaciju

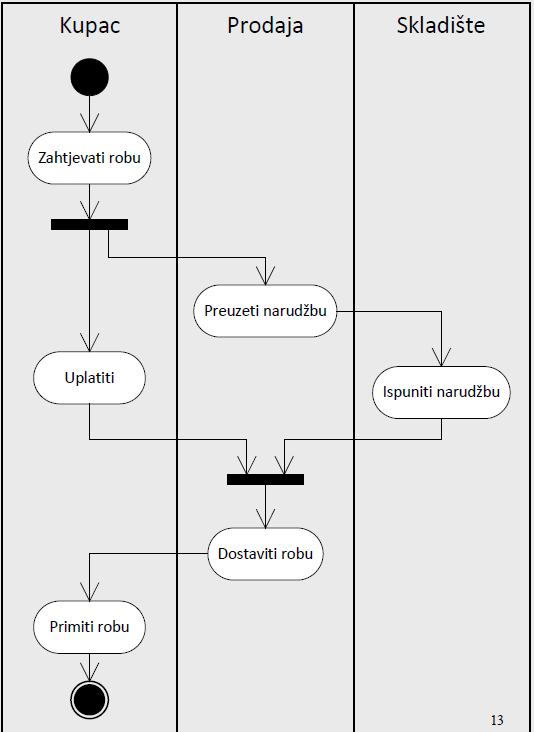


**Dekomponiranje aktivnosti**



**Plivaće staze**

* pokazuje koje se aktivnosti obavljaju i kojim redosljedom ali ne i tok što radi



**CRC kartice**

* modelira odgovornost jednog razreda i njegovu suradnji s drugim razredima
* odgovornosti znanja i radnje

|  |  |  |
| --- | --- | --- |
| **Razred**: Pacijent | **ID**: 3 | **Tip**: Konkretni, Domenski |
| **Opis**: Pojedinac koji treba ili prima medicinsku pomoć | | **Slučajevi korištenja**: 2 |
| **Odgovornosti**   * Zakazati pregled * Odrediti zadnji posjet * Promijeniti status * Dohvatiti povijest bolesti | | **Suradnici**   * Pregled * Povijest bolesti |
| **Atributi:**  Iznos (double)  Nositelj osiguranja (text) | | |
| **Veze:**  **Generalizacija (a-kind-of):** Osoba  **Agregacija (has-parts):** Povijest bolesti  **Ostale veze:** Pregled | | |

**Responsibility driven design** - postupak oblikovanja softvera koji naglašava modeliranje uloga i odgovornosti objekata te suradnje objekata

* **načela**: maksimiziranje apstrakcije, distribucija ponašanja, očuvanje elastičnosti

**Aplikacija** - skup programskih komponenti koje čine logičku cjelinu

* **Logička arhitektura** - razdvajanje tipova funkcionalnosti, broj slojeva >= broj slojeva fizičke arhitekture
* **Fizička arhitektura** - fizička reprezentacija glavnih komponenti IS
* **Petoslojna logička arhitektura**
  + korisničko sučelje - prikaz podataka
  + kontrola sučelja - generiranje izlaza
  + poslovni sloj - poslovna pravila, validacija, obrada, autentifikacija..
  + sloj za pristup podacima
  + sloj za upravljanje pohranom i upravljanje podacima
  + fizička pohrana
* **visoko skalabilni pametni klijent u troslojnoj fizičkoj arhitekturi**
  + DL na aplikacijskom poslužitelju, BL na klijentu i poslužitelju
  + - latencija uslijed mrežnog prometa, smanjenje performansi
  + + velika skalabilnost, preko 1000 istovremenih korisnika

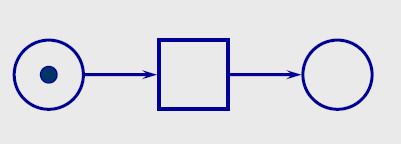
|  |  |
| --- | --- |
| **Arhitekture** | **Aplikacije** |
| Dvoslojne K/S tanki klijenti | Naslijeđeni, vlastiti sustavi gdje je nepraktično i neisplativo odvajanje aplikacijske obrade i upravljanje podacima |
| Dvoslojne K/S debeli klijenti | Aplikacije gdje se aplikacijska obrada izvodi na klijentu COTS programskom podrškom |
| Troslojne ili više slojne K/S | Aplikacije velikog opsega sa stotinama ili tisućama klijenata, sustavi u kojima su i podaci i aplikacije promjenjivi |

**Aplikacijski okvir** - građevni blokovi aplikacije, skup osnovnih softverskih rutina koje čine temeljnu strukturu za razvoj aplikacije u koju se ugrađuju aplikacijski specifične komponente

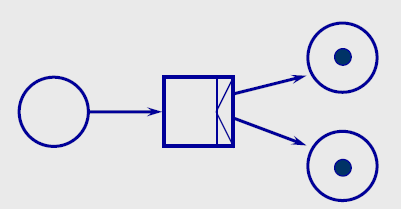
**Enterprise okvir** - cjelovito okruženje za razvoj i ugradnju složenih informacijskih sustava

**Osnovne strukture u definiciji procesa** - poslovni proces, aktivnosti povezane prijelazima, prijelaz

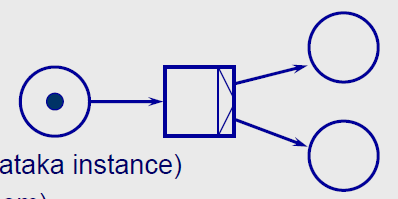
* **4 osnovne vrste prijelaza između aktivnosti** - slijed, odabir, paralelno grananje, ponavljanje



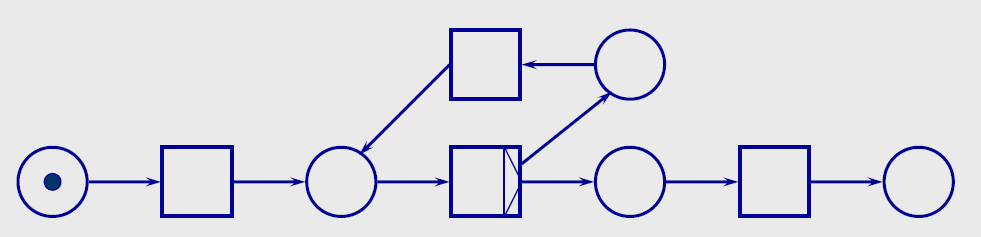
Slijed



Paralelno grananje - moraju se spojiti nazad u jedan



Odabir



Grananje

**Programiranje** - proces pisanja, provjere, ispravljanja pogrešaka te održavanja izvornog programskog koda

**Kodiranje** - pretvorba detaljnog opisa programa u stvarni program, najčešće pisanje izvornog koda nekog formalnog programskog jezika

**Inkrementalno programiranje** - postupno kodiranje i udruživanje razreda i njihovih postupaka

* odrezak (stub) prilikom izrade funkcije koja poziva druge funkcije, pozvane funkcije kodiraju se kao odresci (stub) da tijelo funkcije sadrži poruku ili povratnu vrijednost
* pogonitelj (driver) prilikom izrade funkcije koja će biti pozvana iz neke druge, još neugrađene funkcije, izrađuje se pogonska funkcije (driver)

**Što veća kohezija (prijanjanje modula)** - potrebna visoka unutarnja povezanost elemenata, svaki modul treba obavljati jednu i samo jednu funkciju

**Što manji coupling (vanjska zavisnost modula)** - moduli trebaju biti minimalno međusobno zavisni, minimizacija utjecaja promjene jednog modula na druge module

**Refaktoriranje** - promjena interne strukture programske podrške da bi ju se bolje razumjelo i lakše održavalo uz očuvanje vanjskog ponašanja

* **Lehmanovi zakoni evolucije programske podrške** - povećanje funkcionalnosti razmjerno je smanjenju kvalitete i povećanju složenosti
* + sprječava narušavanje strukture programskog koda, povećanje razumljivosti i čitljivosti programskog koda, olakšano otkrivanje bugova, povećanje produktivnosti
* - pretjerana primjena smanjuje produktivnost i svodi se na puko ref., neautomatizirano refaktoriranje može biti dugotrajno i mukotrpno
* **kada primijeniti**
  + **uvijek** kad god se naprave neke promjene na programskom kodu koje kvare struktutu koda ili kada se promjene očekuju pa se želimo pripremiti
  + **the rule of three**
    - kad 1. put nešto radimo, onda to samo napravimo
    - 2. put u sličnoj situaciji bojimo se dupliciranja ali ga napravimo
    - 3. put u trećoj sličnoj situaciji refaktoriramo
  + **kada se dodaje nova funkcionalnost**
  + **kada treba ispraviti neku bug**
  + **kada se obavlja pregled koda**
  + **bad code smells**

**Testiranje** - provjera programa izvođenjem, uz uporabu ispitnih podataka te analizom rezultata obrade, cilje testiranja je otkrivanje pograšaka odnosno nedostataka unutar programa

* **verifikacija** - dokazivanje da je faza dobro provedena ili da je proizvod dobro napravljen, odgovara specifikaciji zahtjeva
* **validacija** - potvrda kojom se utvrđuje da je napravljen pravi proizvod koji odgovara namjeni te da je prihvatljiv korisniku
* **test** - provjerava je li neki aspekt softvera ispravan
* **pogreška** - propust programera
* **kvar** - neispravan dio koda
* **zastoj u radu** - stanje izazvano jednim ili više kvarova
* **jedinično testiranje** - najmanja jedinica mjere je razred
* **funkcionalno (black boxing)** - provjerava se što cjelina radi i da li zadovoljava zahtjeve, osnovica su CRC, dijagrami razreda, ugovori
* **strukturalno (white box)** - provjerava kako cjelina radi, probni slučajevi iz uvida u programski kod, provode programeri, osnovica su specifikacije metoda
* **integracijsko testiranje** - jedinica je komponenta, razredi koji tvore logičku cjelinu, paket, knjižica, ispitivanje provodi tim
* **testiranje korisničkog sučelja, slučajeva korištenja, interakcije, sučelja sustava, provjera sustava, zahtjeva, uporabivosti, sigurnosti, performansi, dokumentacije**
* **Alfa test** - probna uporaba koju provode korisnici kod izvođača, simulacija stvarnog okruženja
* **Beta test** - provode korisnici kod sebe, bez nazočnosti izvođača
* **Audit test** - potvrda da je sustava gotov, ispravan i spreman za primjenu
* **Test driven development** - testovi se pišu prije koda, u svakoj iteraciji po obavljanju testa provodi se refaktoriranje
  + **regresijsko testiranje** - provjera kojom se dokazuje da softver nije nazadovao, pokretanjem svih testova pri promjeni softvera

**Debugiranje (otklanjanje pogrešaka)** - posljedica testiranja, ali je nezavisna aktivnost, prema simptomima tražimo uzroke

* **tehnike otklanjanja pogrešaka**
  + **brute** **force** - ispisi memorije, praćenje izvođenja koda
  + **backtracking** - početi od mjesta greške pa unazad ići
  + **elimiacija** **uzroka** - izrada skupa ulaznih podataka koji potvrđuju ili opovrgavaju određenu teoriju
* **Klasična zabluda - testiranje je luksuz pred rok završetka**
* **Pravilo 80:20** - 20% modula sadrži 80% svih pogrešaka

**Sistemska** **dokumentacija** - dokumetira razvoj i proizvode pojedinih faza, namijenjena tehničkom osoblju, potreba na razumijevanje, izradu i održavanje sustava

**Korisnička** **dokumentacija** - pomoć korisnicima pri korištenju aplikacija, mora biti prilagođena korisnicima različitih sustava

**Tehnička konverzija** tehnički proces u kojem novi sustav zamijenjuje stari

**Način konverzije**

* **izravno** - prestanak starog i rad novog sustava istovremeno
* **paralelno** - istovremeni rad starog i novog sustava dok se novi ne pokaže ispravan

**Lokacija konverzije** - dijelovi organizacije smješteni na različitim zemljopisnim lokacijama

* **probno** **uvođenje** - izravno/paralelno uvođenje sustava na jednoj lokaciji
* **postupno** - uvođenje slijedno po grupama lokacija
* **istovremeno** - uvođenje istovremeno prema svim lokacijama

**Modularnost konverzije**

* **uvođenje cijelog sustava** - čitav sustav instalira se odjednom (ERP)
* **modularno uvođenje** - postupna zamjena starog sustava s novim, problem što dijelovi moraju biti kompatibilni sa starim sustavom

**Change management** - podrška procesa prilagodbe korisnika na novi sustav

* **ključne uloge** - sponzor promjena (osoba koja želi promjenu), agent promjena (osoba koja uvodi promjenu), usvojitelj promjena (osoba na koju se promjena odnosi)

**Poduka - kako obaviti svoj posao NE KAKO KORISTITI APLIKACIJE!**

* **Načini poduke** - tečajevi, probni rad, prikladna dokumentacija, unutarnji izvođači, vanjski izvođači, potpora tijekom primjene

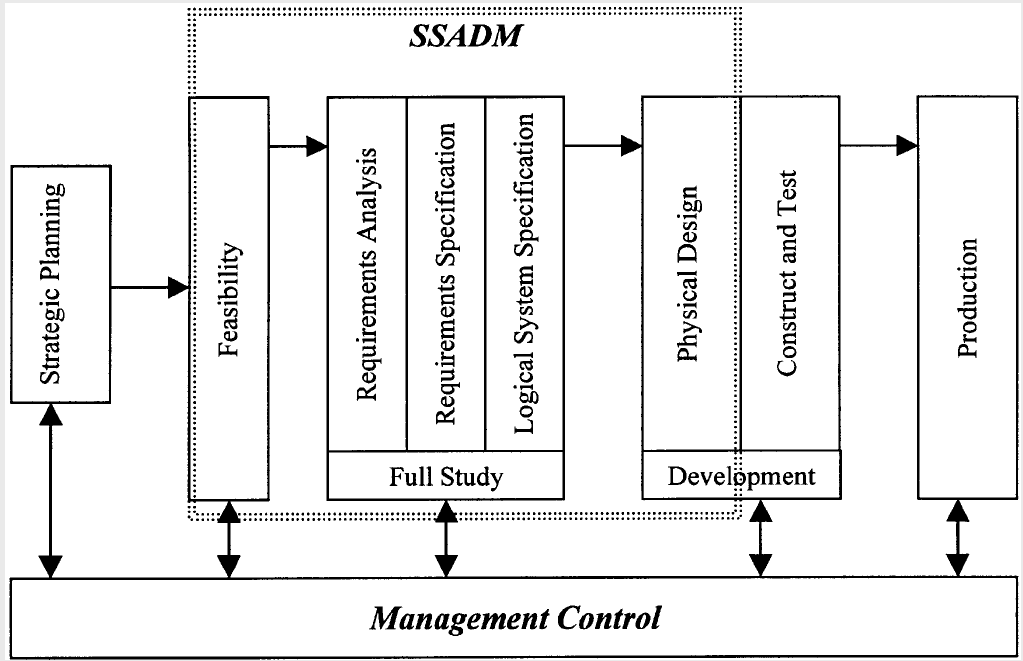
**Održavanje** - sistemska potpora (nakon uvođenja sustav preuzima služba informacijske potpore ili centar komepetencija)

**Višerazinska organizacija podrške**

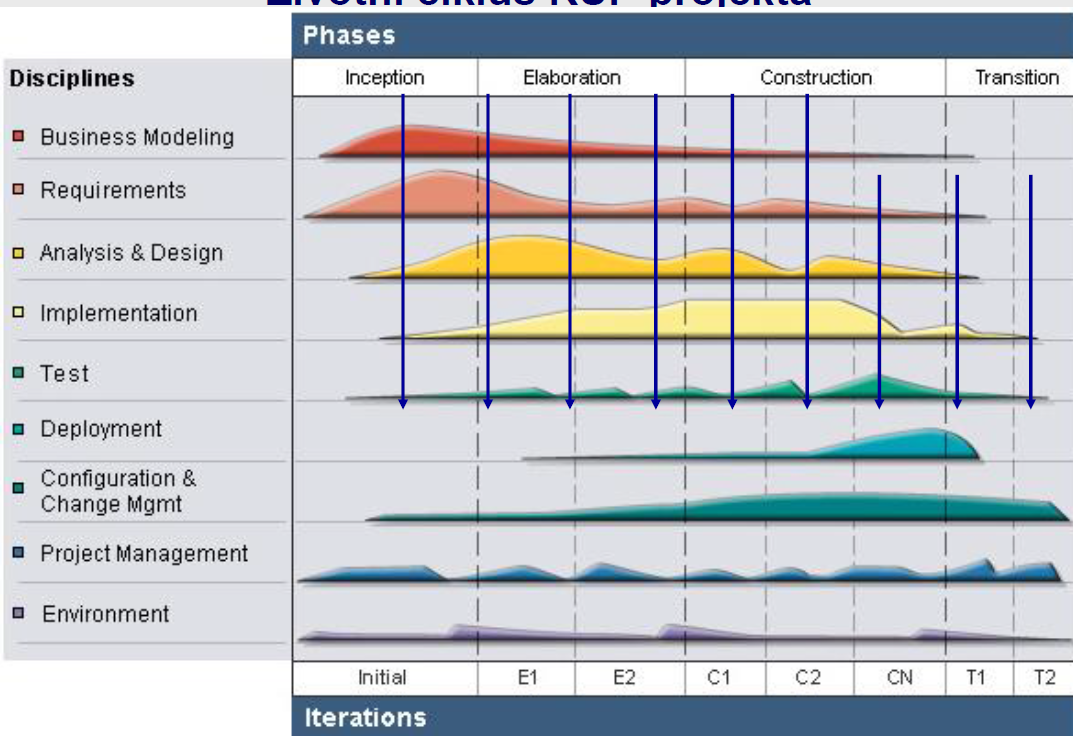
* **1st level support** - odgovara na široki raspon zahtjeva, očekuje se da riješi 80% problema
* **2nd level support** - stručnija pomoć organizirana u timove po struci (mreža, AD, exchange)

**Metodologija** - kolekcija procedura, tehnika, alata i dokumentacijskih pomagala, potkrijepljenih filozofijom, koji potpomažu izgradnju IS

* **komponente** - definirane faze, procedure, dokumentacija, alati, upute
* **vrste**
  + **čvrste (teške)** - zbog opsežnosti, složenosti i velike količine dokumentacije (SSADM, CASE, IEM, RUP)
  + **meke (lagane)** - naglasak na socološkim i ljudskim aspektima, metode za brzi razvoj aplikacija (XP, FDD, SCRUM)
  + **hibridne**
* **karakteristike**
  + **kuhanje po kuharici ne znači da će jelo biti dobro, nedostatci metodologija**
* **Strukturirane metodologije (SSADM)**
  + **faze** - studija izvedivosti, analiza zahtijeva, specifikacija zahtjeva, specifikacije logičkog sustava, fizički dizajn



* **Metodologija informacijskog inženjerstva (IEM)**
  + **razvojni** **okvir** **s 4 stadija** - planiranje, analiza, dizajn, konstrukcija
  + **osnovno** **načelo** - IS treba graditi kao što se grade unikatni proizvodi npr. u graditeljstvu ili brodogradnji
  + **faze** - planiranje strategije IS, analiza poslovnih područja
* **RUP (UDP)**
  + **3 komponente** - skup filozofija, ključnih praksi, načela za uspješan razvoj programske potpore
  + **karakteristike** - produktivnost ekipe (upute, predlošci, baza znanja, svi članovi dijele isti jezik, proces i pristup razvoju), vizualno modeliranje, oslonac na UML, podržanost alatima, iterativan i inkrementalan razvoj, elastičnost i prilagodljivost
  + **načela** - definirati viziju, upravljati prema planu, ublažiti rizike i pratiti probleme, istražiti poslovni slučaj, oblikovati arhitekturu zasnovanu na komponentama, inkrementalno graditi i testirati proizvod, redovito procjenjivati rezultate, upravljati izmjenama i kontrolirati izmjene, isporučiti iskoristiv proizvod, osigurati proces koji odgovara projektu



* **glavne faze razvoja** - **inception** (prikupljanje najvažnijih zahtjeva, procjena ukupnog troška, vremena, rizika, priprema okruženja za projekt), **elaboracija** (prikupljanje detaljnih zahtjeva, definiranje, validacija i zacrtavanje arhitekture, dorada arhitekture i odabir komponenti), **konstrukcija** (provjerava da li su programska podrška, lokacije i korisnici spremni za beta isporuku), **prijelaz** (izvršavanje planova uvođenje u primjenu, omogućavanje proizvoda dostupnim svim krajnjim korisnicima)
* **strategije**
  + **inkrementalna** - poznata domena problema, vrlo dobro poznati rizici
  + **evolucijska** - nova ili nepoznata domena problema, neiskusan tim
  + **strategija** **inkrementalne** **isporuke** - poznata domena problema, iskusan projektni tim, isporuke vrlo važne i imaju visoku vrijednost za korisnika
  + **strategija velikog oblikovanja** - dodaje se inkrement dobro definirane funkcionalnosti na vrlo stabila proizvod
* **Agilni razvoji (XP)** - pojedinci i interakcije važniji od procesa i alata, softver koji radi važniji od dokumentacije, odziv na promjenu važniji od praćenja plana
  + **faze** - **istraživanje** (korisnici bilježe svoje priče na kartice, radi se prororip sustava za testiranje tehnologije i varijanti arhitekture - nekoliko tjedana do mjeseci), **planiranje** (postavlja prioritete na korisničke priče, doseg prvog malog izdanja i vrijeme za pojedinu karticu - nekoliko dana), **iteracije do izdanja** (nekoliko iteracija prije prvog izdanja, klijent određuje kartice koje će se koristiti pri svakoj narednoj iteraciji - jedan do 4 tjedna), **produkcija** (dodatno testiranje i provjera performanski sustava prije isporuke klijentu, razjrješenje primjedbi te odličivanje da li će se riješiti u tekućem izdanju - 3-7 dana), **održavanje** (nakon što je prvo izdanje u produkciji), faza smrti je blizu kada klijent nema više kartica
  + **temeljne vrijednosti** - komunikacija, jednostavnost, povrate informacije, hrabrost, uvažavanje
  + **osnovne prakse** - **priče** (kratki opis funkcionalnosti), **tjedni** **ciklusi** (razvoj u tjednim ciklusima), **kvartalni** **ciklusi** (grublje na dulje staze), **rezerva** (izbjegavanje obećanja koja ne mogu biti ispunjena), **smještaj** **ekipe** (isto mjest za povećanje komunikacije), **cjelovitost** i **zajedništvo** **ekipe** (ekipa sastavljena od članova potrebne stručnosti), **informativno** **radno** **okruženje**, **energičan rad** (programeri moraju biti odmorni), **programiranje u paru** (jedan piše drugi refactor i revizija), **inkrementalni dizajn** (u malim koracima dizajn), **test prije programiranja**, **desetminutna gradnja**, **kontinuirana integracija**
* **SCRUM** - jednostavni upravljački frame za inkrementalni razvoj
  + **uloge** - **ekipa** (7+-2 člana), **vlasnik proizvoda** (zadužen za planove, prioritete, troškove i povrat investicije), **majstor** (brine o procesu, koordinira ali ne donosi poslovne ni tehničke odluke), **klijent**, **uprava**
  + **životni ciklus** - **prije** **igre** (radna lista proizvoda se radi, planiranje i dizajn), **razvoj/igra** (razvoj iterativnim ciklusima ili sprintovima, trajanje oko 30 dana pojedinog sprinta 7-21 u praksi, 3-8 sprintova dok sustav ne bude spreman za distribuciju), **poslije igre** (priprema sustava za izdanje kroz integraciju, testiranje i druge aktivnosti)
  + **artefakti** - **product backlog** (neizvršen rad, preostali posao vidljiv svim dionicima), **sprint** **backlog** (popis zadataka i statusa vidljiv ekipi), **product backlog item** (definira što najčešće kao korisnička priča, sadrži više zadataka), **zadatak** **sprinta** (kako za PBI što)
* **CASE** - automatizacija programske opreme, programski sustav za pomoć pri izgradnji IS i razvoju programske opreme
  + **vrste** - **gornji** (rane faze), **donji** (izrada, ugradnja), **integrirani** (cijeli ciklus)
  + **preoblikovanje programske podrške** - skup ispitivanja i izmjena postojećeg sustava koji rezultiraju preustrojenjem, mijenjanje čitavog sustava ili malog dijela
  + **inženjerstvo prema naprijed** - s viših razina apstrakcije na projektiranje i izvođenje
  + **inženjerstvo unatrag** - prepoznavanje dijelova sustava i njihove povezanosti s ciljem da se identificiraju komponente sustava i dobije potpuna specifikacija, izvorni kod dostupan ali je dokumentacija nepotpuna neažurna ili je nema

**sustavi za upravljanje poduzećem**

* **ERP** - integrirani informacijski sustava za upravljanje poslovanjem, podržava financijsko poslovanje, proizvodnju, robo-materijalno poslovanje, upravljanje ljudskim resursima i plaće, ne mora nužno unaprijediti sustav, cijena ovisna o tržištu namijenjen više velikim tvrtkama kojih je trneutno malo na tržištu
  + **upravljanje odnosom s kupcima** - posljedica poslovne filozofije usmjerena na što bolje upoznavanje kupaca kako bi se što bolje zadovoljile njihove potrebe i želje, saznati što više o kupcu i onda mu to pružiti
  + **upravljanje lancem nabave** - podrška procesim planiranja i provedbe protoka materijala i proizvoda, planiranje potreažnje za gotovim proizvodima, odabir najboljeg dobavljača, kada poduzeća koriste internet postoji mogućnost povezivanja materijalnih tokova poduzeća
  + **implementacija** - **evaluacija** (određivanje najvažnijih zahtjeva), implementacija (2 faze), **proširenja**, **održavanje**, **opadanje** **vrijednosti** (promjena poslovanja s vremenom održavanje sve teže)
  + **problemi uvođenja** - vremenski dugotrajan proces, trošak (veliki početni trošak, neočekivani skrivani troškovi, konzultant na dan iznad prosječna plaća), ovisnost o proizvođaču, potreba za kvalificiranim osobljem

**Upravljanje projektom**

* **projekt** - vremenski određeno nastojanje da se proizvede jedinstven proizvod, usliga ili rezultat, svaki mora imati jasno određen početak i kraj, završava u trenutku kada su ciljevi projekta dostignuti ili kada se zaključi da ciljevi projekta ne mogu ili neće biti dostignuti
* **operacije** - neprekidne i mogu se ponavljati, svrha je podupiranje i održanje poslovanja čak i kada se ciljevi promijene
* **podprojekti** - projekti se često dijele na podprojekte koji su upravljiviji
* **upravljanje** - primjena znanja, vještina, alata i tehnika u projektnim aktivnostima da bi se ispunili projektni zahtjevi (plan, sredstva, organizacija, vremenski raspored, upravljanje, kontrola, nadzor)
* **procesi** - započinjanje, planiranje, izvršavanje, nadzor i kontrola, zatvaranje

**Organizacija projekata**

* **ekipni** **rad** - grupa međuzavisnih pojedinaca koji zajedno rade na ostvarivanju zajedničkog cilja te dijele odgovornost za završetak posla, kvalitetnije donošenje odluka, motivacija članova, inovativnost
* **razvoj ekipe** - formiranje, jurišanje, normiranje, predstavljanje
* **Ekipa s glavnim programerom**
  + glavni programer utjelovljuje znanje i odlučivanje ekipe mora istovremeno biti dobar programer i voditelj, rezervni programer služi kao zamjena za nekog od mlađih programera
* **4GL ekipa**
  + voditelj projekta VSA, suradnja s korisnikom PA, konceptualno i logičko oblikovanje SA, isporuka sustava PA, nabava i pogon opreme SIZR, mrežni servisi SIZK, izrada dokumentacije, potporno osoblje
* **XP ekipa**
  + razvojnik, klijent, tester, pratilac (prati procjene ekipe), trener (vodi ekipu kroz proces), savjetnik (vanjski član, tehničko znanje o projektu), upravitelj (šef)
* **prilagodba strukture tima**
  + **elastični model** - upravitelj, voditelj, projektant, programer, DB admin, sistem inženjer
* **organizacija velikih projekata**
  + upravitelj projekta, upravitelj ekipe, voditelj ekipe, ekipa
* **karakteristike dobre ekipe** - međusobno povjerenje, učinkovita komunikacija, osnaživanje, mali broj članova 7+-2, uživanje u poslu, zajednička vizija, kompetentni članovi, predanost timu
* **karakteristike dobrog voditelja** - mora zastupati interese organizacije i interese članova tima, mora biti podržan u provođenju svojih ovlasti inače gubi autoritet, objektivnost, mogućnost nagrađivanja i kažnjavanja tima, tehnička kompetencija, brzo i efikasno rješavanje konflikata
* **vrednovanje** - državne institucije (koeficijenti), poduzeća (plaće po učinku), dohodak od nesamostalnog rada (honorar)

**Elementi upravljanja projektom**

* **planiranje projekta** - odrediti doseg, troškove, voditelje, upravitelje, pokrenuti projekt, veličina projekta, funkcijske točke, napor izrade, vrijeme izrade
* **vremenski raspored** - projekt kao definiran redoslije aktivnosti, planiranje vremena, izrada rasporeda
  + **kritični put** - niz zadataka koji moraju završiti na vrijeme da bi projekt završio na vrijeme, svaki zadatak na tom putu je kritični
* **upravljanje** **komunikacijom** - zajedničko dnevno druženje svih članova ekipe, dnevno po 15min, po potrebi i izvanredni sastanci
* **upravljanje** **vremenom** - raspoređivanje poslova, izbjegavanje obavljanja tuđih oslova, neprimjeren utrošak vremena (druženje, telefonski..)
* **dokumentiranje** - povelja projekta, plan upravljanja projektom

**Mitovi i legende** - zašto planirati?, neuspješno planiranje = planiranje neuspjeha

* **paret** - 20% problema izaziva 80% posla
* **parkinson** - rad se povećava tako da troši čitavo planirano ili raspoloživo vrijeme
* **brooks** - dodavanje osoblja u projekt koji kasni povećava kašnjenje
* **fertalj** - za neki posao treba vremena upravo onoliko koliko je raspoloživo