

# Modeliranje Ciljnog Sistema

Dragutin Ilić

1100/2015

# Uvod

- Razvoj softvera: proces ili sistem koji se sastoji od kreiranja, analize i upotrebe serije modela koji se mogu preklapati u vremenu
- Model životnog ciklusa:
  - Platforma za planiranje, raspoređivanje i kontrolisanje projekta
  - Nedovoljna za kreiranje softvera
  - Potreban drugi tip modela

# Sistemiški model

- Izlaže celokupnu arhitekturu, odnose i operacije cilnog sistema
- Opisuje šta će razvojni tim izgraditi
- Oslikava sekvencijalne i paralelne aktivnosti u okviru životnog ciklusa
- Model se bazira na rezultatima prethodnih modela i nekih novih informacija
- Komplementaran sa drugim modelima
- Nedovoljan za obavljanje celokupnog posla
- Koriste se različite metode sistemskog modeliranja u kombinaciji sa modelom životnog ciklusa

# Zašto modelirati softverske sisteme?

- Manja je cena pronalaženja problema prilikom izgradnje modela sistema i njihove analize nego nakon izgradnje samog sistema.
- Klasična greška menadžera: „počni sa kodiranjem što ranije“

Faza životnog ciklusa	Minimalna cena da se ispravi greška	Tipična cena da se ispravi greška	Maksimalna cena da se ispravi greška
Definisanje i dizajn zahteva	1	1	1
Kodiranje i testiranje	5	10	20
Instalacija i upotreba na terenu	25	100	Ekstremna cena u slučaju gubitka života

# Zahtevi metoda modeliranja

- Reprezentuju ugovor između snabdevača softvera i entiteta koje finansira njegov razvoj
- Često se izražavaju legalističkim stilom koji je nedovoljan
- Potrebni su opisni alati koji daju učesnicima u razvoju bolji uvid u izgled i funkcionalnost sistema i na koji način će ispuniti zahteve
- Alati:
  - Dijagram toka podataka
  - IDEF0
  - Matrice Samo-Interakcije

# Dijagrami toka podataka - DTP

- Jednostavno se prevode u neku formu dizajna
- Grafički iskazuje zahteve
- Pruža sredstva da klijent i softverski tim mogu da kritikuju i „debuguju“ grafički predstavljen postojeći sistem (ukoliko postoji) i grafički predstavljen budući sistem
- Smanjuje šansu da promašimo zahteve u isporučenom sistemu

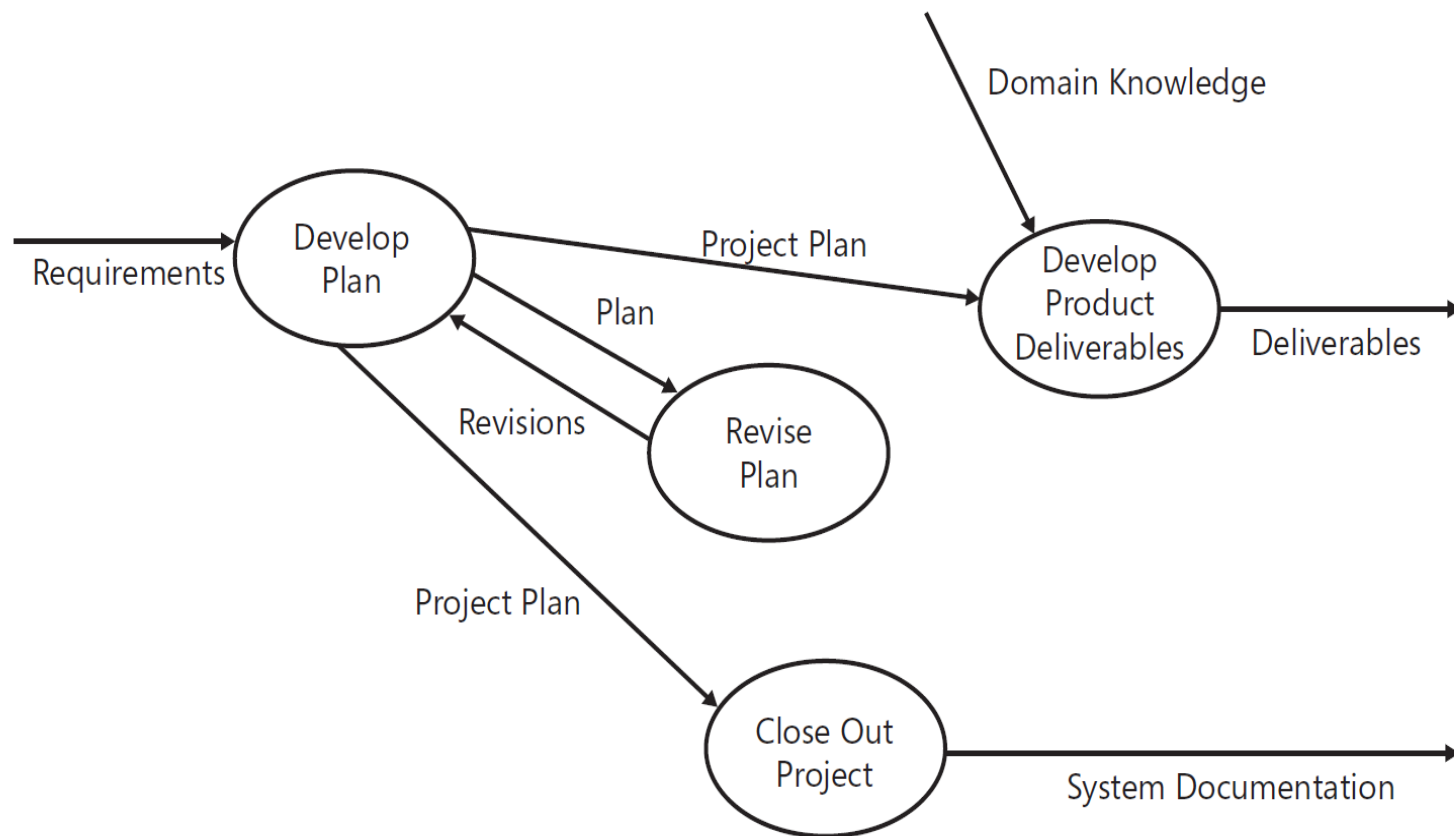
# Dijagrami toka podataka – DTP

- Komponente DTP-a:
  - Elementi procesa
  - Elementi toka podataka
  - Elementi skladištenja podataka
  - Izvorni i ishodni elementi
  - Rečnici podataka

Obično se koriste hijerarhijski ili višenivoini DTP:

- U prvom nivou jedan osnovni, opšti proces
- Na narednim nivoima se proces dekomponuje u više detaljnijih procesa

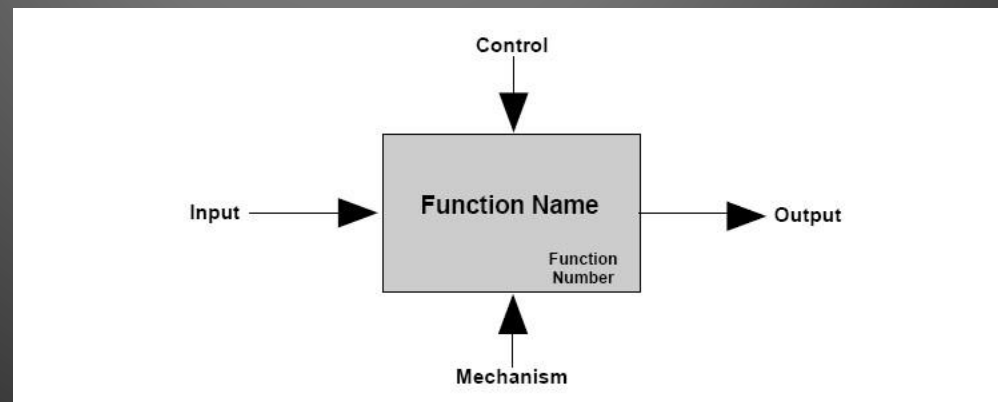
# Dijagrami toka podataka – DTP



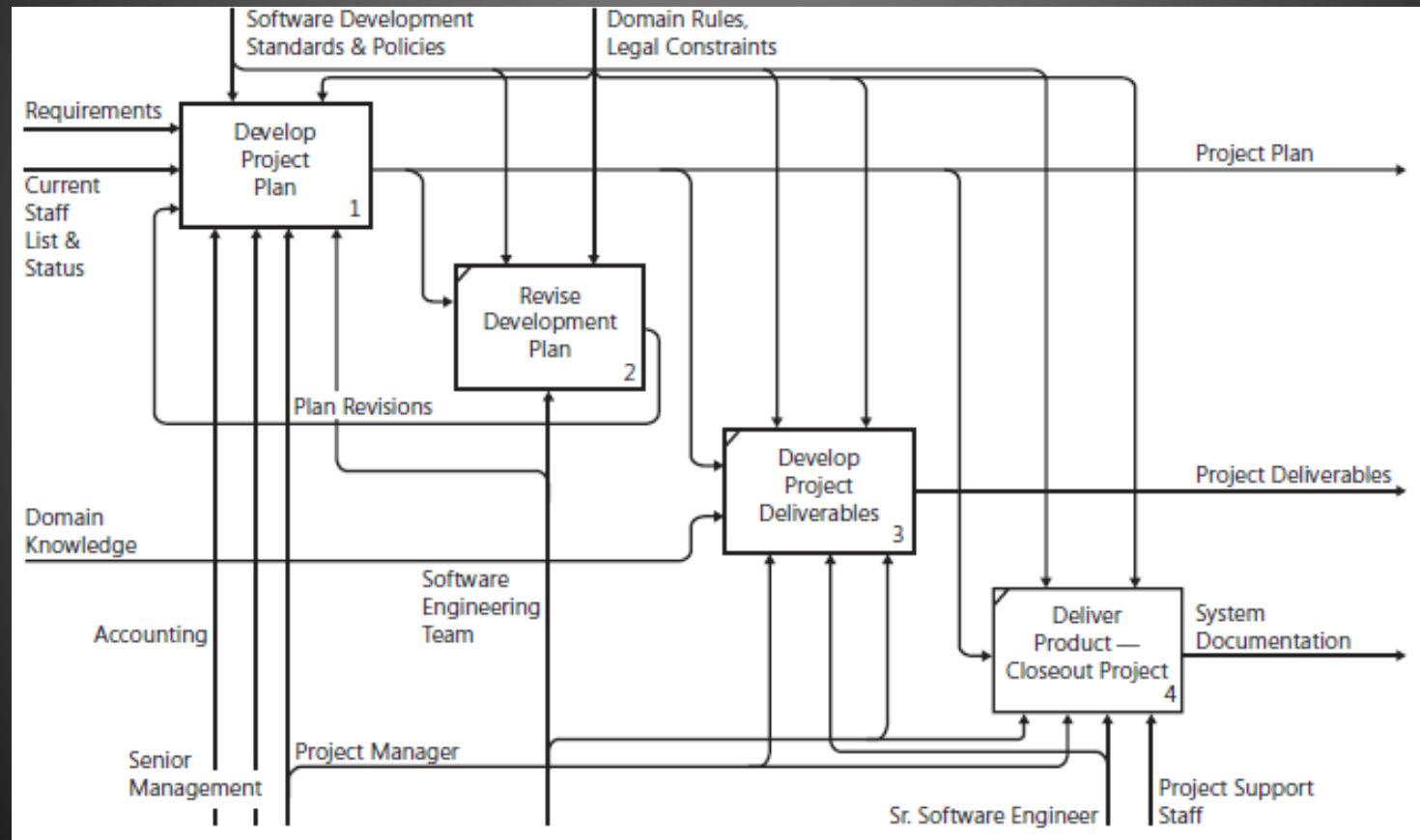


# IDEF0

- Detaljniji od DTP-a
- Fokus je na važnim detaljima kao što su *mehanizmi* (ko ili šta će postići transformaciju) i *kontrole* (pravila ili smernice sa kojima se sistem mora uskladiti)
- Sastoji se od ćelija aktivnosti (funkcije), ulaza, izlaza i gore pomenuta dva elementa



# IDEF0 - Primer



# Analiza zahteva korišćenjem matrica samointerakcije

- U jednom trenutku će tekstualna, legalna verzija liste zahteva biti proizvedena na osnovu koje će moći da se utvrdi (preko ček liste) da li softver obavlja sve ono što je nameravano.
- Malo menadžera prepoznaje važnost u obavljanju analize zahteva
- Ukoliko se ne prepozna zavisnost nekih zahteva to može uticati na kašnjenja i dodatnu preradu
- Analizom vršimo sinhronizaciju i na taj način poboljšavamo produktivnost razvojnog tima
- *Analiza* = ispitivanje, usavršavanje i/ili reorganizacija

# Analiza zahteva korišćenjem matrica samointerakcije

- Analiza odnosa izmedju zahteva je znacajna zato što dokumentuje pozitivni ili negativni uticaj promene nekog zahteva na druge zahteve
- Analiza odnosa se obavlja matricom samointerakcije
  - $n \times n$  matrica
  - $n$  – broj zahteva
  - Glavna dijagonala se ignoriše jer se odnosi na taj zhtev
  - Iznad glavne dijagonale se matrica popunjava znacima + ili – u zavisnosti postojanja odnosa izmedju zahteva
  - Mogu se javiti i oznake *I* (*inhibits*), *E* (*enhances*) i *NI* (*no impact*)

# Analiza zahteva korišćenjem matrica samointerakcije - primer

[illegible]

# Specifikacioni metod tabele odziva sistema - SRT i sistemi u realnom vremenu

- Kompaktnija i preciznija sredstva za uspostavljanje zahteva pri razvoju softvera koji se koristi za ugradjene sisteme u realnom vremenu
- Klijent specifikuje zahteve koji se očekuju od novog sistema (*Odziv*) kada se određeni ulazi ili događaji (*Stimulusi*) pojave
- *SRT daje testerima rani uvid kako treba da se testovi kreiraju kako bi se osiguralo da sistem radi šta se od njega očekuje*
- *Zahtevi se navode kao serija kratkih izjava*

# Specifikacioni metod tabele odziva sistema - SRT i sistemi u realnom vremenu

- Definicije (karakteristike) sistema u realnom vremenu:
  - brz odziv
  - konkurentnost
  - kompleksna sredina
  - kritično vreme
- Problem sistema u realnom vremenu je njegova grafička, tekstualna ili tabelarna reprezentacija koja će biti isporučena klijentu.
- Jedno od rešenja su Stimulativni-Odzivni scenariji (SRS)



# Stimulativni-Odzivni scenariji (SRS)

- Sistem = „Crna kutija“
- Serijom diskusija i intervju a klijent izlaže koji scenarije očekuje od „Crne kutije“ kada se javi određeni stimulus
- Scenariju se dodeli ime i svi njegovi stimulusi i odzivi su opisani u formularu
- C = uslov, S = stanje i D = podatak - kao podkategorije za stimulus i odziv

Stimulus	Step in Sequence	Response
C—User turns on system. S—System power off. D—User enters ID and password	1. Power on.	D—Display Power On light. C—Initiate boot sequence. S—Power on.
S—System access disabled.	2. Logon.	C—Valid logon. D—Confirm to user. S—System access enabled.



# Slučajevi upotrebe

- Popularno sredstvo za predstavljanje zahteva komunicirajući sa mušterijom i razvijaoцем istovremeno
- Opisani su dijagramima i pratećim tekstom
- Koriste se u interaktivnim sistemima (interakcija između učesnika i sistema ili između sistema sa drugim sistemom)
- Ne postoji standardan način predstavljanja dijagrama
- Ne postoji odnos slučajeva upotrebe sa dizjanom sistema

# Slučajevi upotrebe

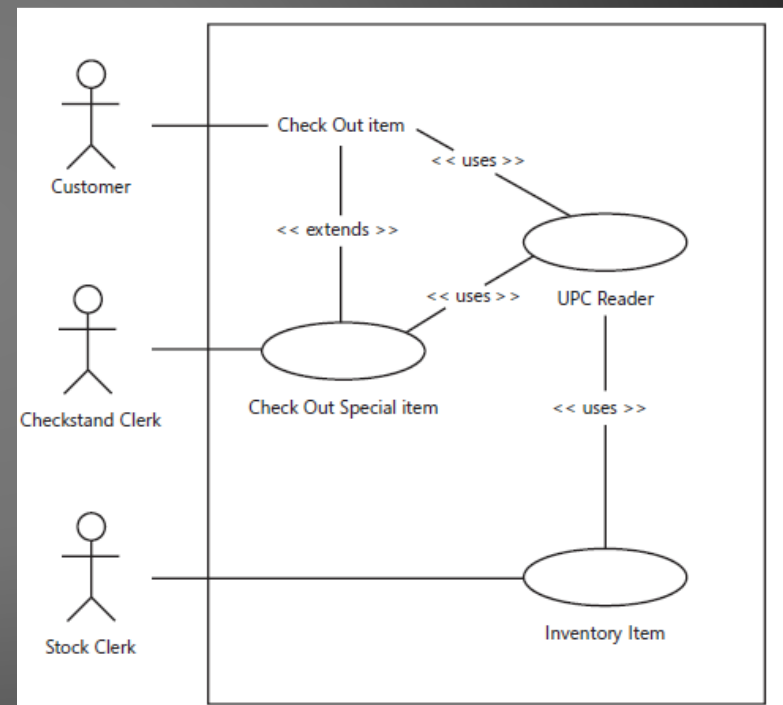
- Mušterjija vrši kupovinu u prodavnici koristeći samouslužni sistem:

1. Customer presses Start button.
2. System displays instructions.
3. Customer scans each item, touches it to the yellow pad to the right of the scanner, and then places the item in the bag provided.
4. System displays item description, price, and current total.
5. Customer presses Pay Now button.
6. System display Methods Of Payment options screen.
7. Customer selects method of payment.
8. Customer pays, takes receipt, and leaves.

- Neki mogući slučajevi greške:

- Item bar code cannot be scanned—system emits error tone summoning clerk for assistance.
- Item (for example, fruit) has no bar code to scan—customer is prompted to manually enter code from chart provided.
- Customer enters invalid item code—system emits error tone to summon clerk for assistance.

- Dijagram opisanog scenarija:



# Pregled metoda dizajna

- U početku Fon Nojmanovi dijagrami toka (bolji za dokumentovanje programa nakon što je napisan)
- Veliki broj dizajna od raznih kompanija i individua
- Pomažu razvijaocima softvera u izražavanju njihovih ideja koje se tiču sistema
- Dizajn mora imati veze sa:
  - Arhitekturom
  - Odnosima
  - Korisničkim interfejsom

# Pregled metoda dizajna

- Postoji više šema za klasifikaciju metoda dizajna sistema
- Jedna od korisnijih je šema sa četiri kategorije:
  - orjentisan ka podacima
  - funkcionalno orjentisan
  - objektno orjentisan
  - formalan

# Odabir odgovarajuće tehnike za modelovanje sistema

- Moramo znati da je svaki metod „samo alat u kutiji alata“
- Metod je kreiran da odgovori na odgovarajući zadatak i naglasi odgovarajući domen informacija
- Ne postoji idealan metod modelovanja.
- Zadatak projektnog menadžera je da osigura da metode modelovanja koje će se koristiti na projektu budu:
  - odgovarajuće
  - dokumentovane
  - odobravajuće od strane razvojnog tima

# Zaključak

- Svaka tehnika modelovanja reflektuje specifične i nekad jedinstvene tačke gledišta njihovih autora
- Često razvojni timovi nastoje da koriste metode sa kojima su upoznati pri tome ne analizirajući da li postoji neka bolja metoda
- Dosta metoda se koriste zato što su popularizovane od strane tehničkih konferencija i izdavača, a ne zato što su najbolji izbor
- Treba imati u vidu da ne postoji metod modelovanja sistema koji je najbolji u svim situacijama



HVALA NA PAŽNJI!