

01 - Uvod

Sistemske softver - programi niskog nivoa koji omogućavaju rad sa računarskom opremom (hardverom).

Aplikativni softver - programi napravljeni za specifičnu svrhu, prema potrebama korisnika. Nije u direktnoj vezi sa hardverom, već se oslanja na sistemski softver (OS).

Softversko inženjerstvo - inženjerska disciplina koja se bavi svim aspektima proizvodnje softvera.

Metode (ili *tehnike*) - formalne procedure za pravljenje nekog izhoda

Kuvar primenjuje različite tehnike da bi dobio odgovarajući preliv

Alati - instrumenti ili auto. sistemi koji omogućavaju da se nešto obavi na bolji način (*nisu presudni*)

Procedure - kombinacija alata i tehnika koji zajedno proizvode dati proizvod - *Recept*.

Paradigma - poseban pristup ili filozofija gradnje softvera

Stil kuvanja (kineski/francuski) ; Paradigma (proceduralni/oo razvoj)

Sistem - skup objekata i aktivnosti i njihovih međusobnih veza

02 - Modelovanje procesa i životnog ciklusa

Proces - uređeni skup zadataka. Podrazumeva niz koraka koji vode do željenog cilja.

** Koraci procesa obuhvataju aktivnosti, ograničenja i resurse.*

Životni ciklus - proces izrade nekog proizvoda

Kolač sa čokoladnim prelivom

- Proces sadrži više postupaka
 - kupovina sastojaka, ..., pronalaženje odgovarajućeg posuđa
- Recept opisuje postupak za mešenje i pečenje kolača
 - Aktivnosti: ulupati jaje pre mešanja sa ostalim sastojcima
 - Ograničenja: pre mešanja sa šećerom čokolada mora biti ugrejana do tačke topljenja
 - Resurse: šećer, brašno, jaja i čokolada
- Preslatko? Smanjiti količinu šećera u receptu.
- Nema dovoljno čokolade? Dodati kakao.
- Bez recepta (dokumentacije) kuvar ne bi znao kako da napravi izmene i proceni njihove efekte
- Prenosjenje iskustva

Faza - proces ili skup procesa

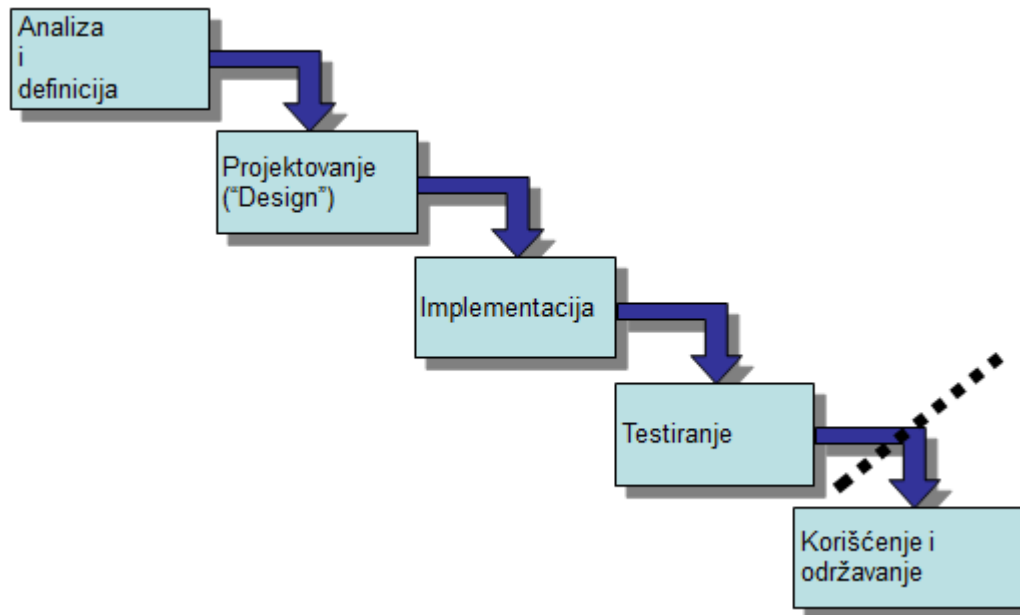
Proces - skup aktivnosti. Aktivnost obuhvata ograničenja, izlaze i resurse.

Model - recept za sprovođenje procesa razvoja softvera

Model - opis načina na koji se razvoj softvera stvarno odvija

Učesnici : Analitičar, projektant, programer, testni inženjer, predavač.

Model vodopada - jedna faza razvoja treba da se u potpunosti okonča pre nego što se započne sl.



Analiza - skupljanje zahteva i analiza šta je potrebno raditi.

Dizajn - konkretan plan kako će biti implementiran sistem.

Implementacija - faza gde se projekat implementira u određenom prog. jez i platformi. *Na kraju spreman za upotrebu.*

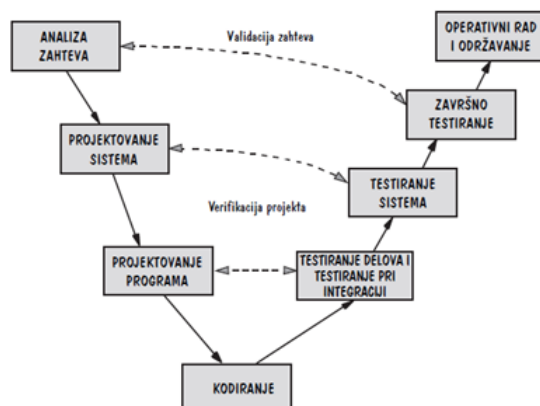
Verifikacija - testira sistem iz prethodne faze, na kraju spreman na predaju krajnim korisnicima.

Održavanje - vršenje eventualnih dorada u skladu sa izmenama traženim od korisnika.

Gantov dijagram - prikazuje koje aktivnosti je potrebno izvršiti, ko će ih izvršiti, kada će početi, kada će se završiti i međusobne zavisnosti među aktivnostima.

Prototip - delimično razvijen proizvod koji omogućava klijentima da ispituju neke aspekte predloženog sistema i odluče da li je on pogodan ili potreban u sklopu gotovog proizvoda.

V-model - kao i vodopad, u sledeću fazu se prelazi samo ako je završena prethodna, ali svaka faza razvoja ima odgovarajuću fazu testiranja kojom se ona validira i posle svake faze pre prelaska u narednu se planira kako će se izvršiti verifikacija te faze (i ako se ona vrši na kraju projekta).



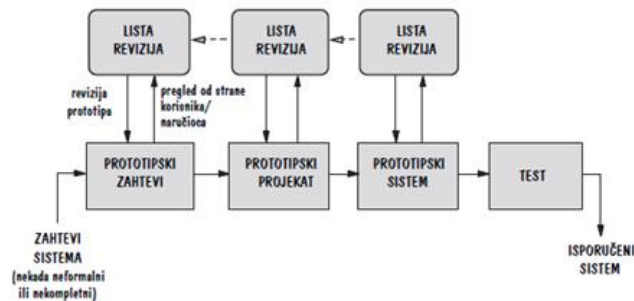
Jedinično testiranje - testiranje pojedinih delova sistema (modul, komponente, forme).

Integraciono testiranje - testiranje komunikacije, povezivanje i tokova među modulima.

Sistemska testiranje - testira sistem u celini.

Test prihvatljivosti - njime krajnji korisnici potvrđuju da aplikacija radi upravo ono što im treba.

Prototipski model - omogućava da se kompletan ili delovi sistema brzo konstruišu radi razjašnjenja ili boljeg razumevanja otvorenih pitanja. Cilj je smanjenje rizika ili neodređenosti prilikom projektovanja.



03 - Ujedinjeni razvojni proces (RUP)

RUP (Rational Unified Process) - metodologija (iterativno-inkrementalnog) razvojnog procesa.

* *Uloge (ko?)* - definiše skup povezanih veština, sposobnosti i odgovornosti

* *Proizvodi rada (šta?)* - predstavljaju rezultat nekog zadatka (modeli, proizvodi, dokumentacije...)

* *Zadaci (kako?)* - opisuju posao dodeljen ulazi koji proizvodi neki koristan rezultat

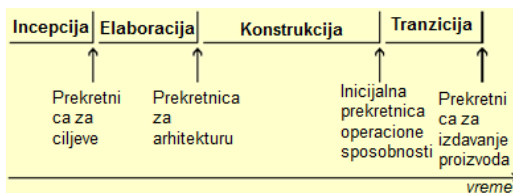
RUP posmatra životni ciklus projekta kroz četiri faze, svaka se završava važnom prekretnicom.

1. Faza započinjanja - razumevanje obima i ciljeva projekta, utvrđivanje šta treba da se napravi, obim sistema...

2. Faza razrade - definisanje osnovne arhitekture sistema na osnovu use case-ova, smanjivanje rizika u pogledu zahteva

3. Faza konstrukcije - razvijanje komponenata, testiranje i kompletiranje dokumentacije projekta.

4. Faza tranzicije - isporučivanje gotovog proizvoda.



Poslovni model - prikaz vizije poslovanja organizacionog sistema za koji se razvija proizvod. Na osnovu njih se specificiraju procesi, uloge i odgovornosti.

Zahtevi - forma iskazivanja potreba svih zainteresovanih strana.

Analiza i dizajn - cilj je formulisanje načina izgradnje sistema tako da izvršava tražene funkcije i ispunjava sve zahteve. Rezultat je Design Model koji služi kao osnova za kodiranje.

Implementacija - cilj realizacija softvera i integracija rešenja poj. prog. u finalni proizvod.

Testiranje - verifikacija rada programa i identifikacija grešaka pre isporuke kupcu.

Uvođenje u eksploataciju - cilj uspešna isporuka i UUE softvera na strani krajnjeg korisnika.

Vizija - Definiše poglede zainteresovanih strana na proizvod koji se razvija, u vidu njihovih zahteva.

Analitički model opisuje realizaciju use case-ova i apstrakcija je **projektnog modela** koji opisuje realizaciju use case-ova i aps. je **implementacionog modela** koji je zbirka komponenata.

04 - Planiranje i upravljanje projektom

Projekat - svaki poduhvat, posao ili proces koji predstavlja objekat planiranja. Skup aktivnosti koje nas dovode do realizacije.

* Razvoj projekta možemo da podelimo na niz faza koje se dele na korake koji se dele na nizove aktivnosti.

Projektni rokovi - opisuju razvojni ciklus nekog projekta, sa fazama od kojih je svaka razbijena na zadatke ili aktivnosti koje je potrebno realizovati.

Aktivnost - deo projekta koji se odigrava tokom nekog vremena. Ima početak, trajanje i završetak.

Događaj (prekretnica) - predstavlja završetak aktivnosti ili projekta u određenom vremenskom trenutku.

Metode mrežnog planiranja - zajednički naziv za postupke planiranja i upravljanja projektima koji imaju zajedničko grafički model toka procesa (tj. mrežni dijagram) koji predstavlja grafički prikaz projekta.

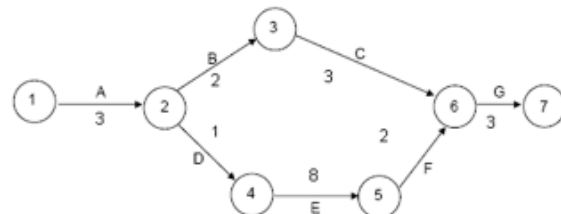
CPM (Critical Path Method) i **PERT (Program Evaluation and Review Technique)** su MMP.

Postoje 3 faze u primeni mrežnog planiranja :

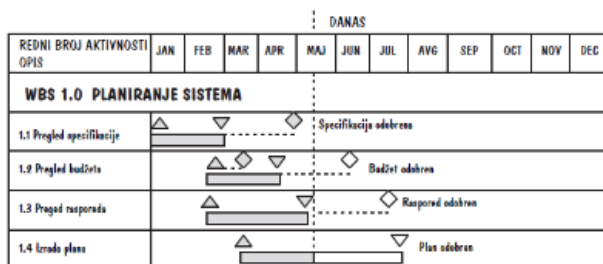
1. **Analiza strukture** - utvrđivanje redosleda i međusobnih zavisnosti projekta.
2. **Analiza vremena** - određivanje minimalnog vremena za izvršenje projekta.
3. **Analiza troškova** - utvrđivanje minimalnih troškova.

Proces se prvo deli na aktivnosti, a zatim se utvrđuje zavisnost među aktivnostima. Popis aktivnosti prikazujemo u obliku tabele međuzavisnosti aktivnosti. Tabela sadrži opis, vreme trajanja i preduslove. Analizu pomoću CPM počinjemo crtanjem grafa aktivnosti na osnovu tabele MĐA. Čvorovi predstavljaju događaje (prekretnice) projekta, a grane koje povezuju čvorove predstavljaju sadržane aktivnosti.

	Aktivnosti	Trajanje (u danima)	Preduslovi
A	Projektovanje IS-a	3	
B	Nabavka računara	2	A
C	Instaliranje računara	3	B
D	Generisanje baze podataka	1	A
E	Izrada aplikacije	8	D
F	Obuka korisnika	2	E
G	Testiranje	3	C,F



Gantogram - metoda planiranja koja omogućava grafičko i vremensko planiranje odvijanja zadataka, odnosno grafički prikaz odvijanja zadataka u vremenu. Na Gantovom dijagram na horizontalnoj osi je naznačeno vreme, a na vertikalnoj osi su poredane aktivnosti.



Rizici mogu biti **vezani za projekat** (razvoj projekta ili resursi), **za proizvod** (kvalitet softvera), **poslovni** (org. koja izvršava ili finansira projekat). 4 faze upr. rizicima : Identifikacija > Analiza > Planiranje > Nadziranje

08 - Testiranje softvera

Metodi "crne" i "bele kutije" predstavljaju dve krajnosti, jer se u prvom slučaju ništa ne zna o procesu obrade podataka, dok se u drugom slučaju zna sve.

Metod crne kutije - program se smatra kao zatvorena kutija nepoznatog sadržaja kod koje su vidljivi samo ulazi i izlazi. Prilikom testiranja na osnovu zadatih ulaza dobijeni izlazi se upoređuju sa unapred očekivanim i tako se proverava ispravnost programa. Prednost što je testiranje oslobođeno brige o ograničenjima koja proističu iz unutrašnje strukture komponente i logike njenog rada. Mana što je detaljno testiranje svih kombinacija različitih ulaza praktično neizvodljivo.

Metod bele kutije - testira i analizira izvorni programski kod. Program se smatra kao otvorena kutija čija je unutrašnjost poznata. Plan testiranja se formira na osnovu strukture programa, tako se mogu generisati testovi koji izvršavaju sve naredbe u programu, pozivaju sve funkcije...može se proveriti celokupan kod. Mana što testiranje ove metode može da bude praktično neizvodljivo, npr. ako sadrži mnogo granjanja i petlji broj putanja može biti ogroman.

Fazni model

- Projekat se pravi tako da se može rastaviti na podsisteme na osnovu funkcija. Postoje dva paralelna sistema : razvojni kojim se bave developeri i produkioni koji korisnici trenutno koriste. Kod iterativnog dizajna, korisnicima se isporučuje sistem u potpunosti i na njegovim podsistemima se vrše dorade. Kod inkrementalnog dizajna, korisnicima je isporučen deo funkcionalnosti na koju se dodaju nove funkcionalnosti (podsistemi).