. Pojam sistema, informacionog sistema, podatak i informacija, upravljanje.

Sistem je skup objekata i njihovih veza.

"Informacija je kapacitet povećanja znanja" - I.Wilson

"Informacija je nešto što ukida ili smanjuje neodredjenost" - N.Winer.

Informacioni sistem je sistem u kome se veze izmedju objekata i veze sistema sa okolinom ostvaruju razmenom informacija.

Informacioni sistemi su sastavni deo upravljanja nekog sistema.

Definicija: Upravljački informacioni sistem je sistem koji prenosi, čuva i obradjuje podatke u informacije potrebne za upravljanje.

Podatak je kodirana predstava o nekoj činjenici iz realnog sveta.

Informacija je protumačeni podatak o pojavi koju podatak prikazuje.

2. Klasični sistem datoteka i pojam baza podataka.

Redundansa podataka je višestruko pamćenje istih podataka. Kada se neki podatak promeni, to se mora učiniti na svim mestima na kojima se on čuva. To povećava troškove obrade podataka, i u raznim izveštajima, pojavljuju se različite verzije istog podataka.

Zavisnost programa od organizacije podataka. Programi su zavisni i od logičke i od fizičke strukture podataka. Novi zahtevi za informacijama zahtevaju izmene u ranije razvijenim programima. Umesto toga, obično se za novi program stvara nova datoteka, a to dalje povećava redundansu podataka.

Niska produktivnost u razvoju IS. Struktuiranje podataka u nezavisan skup datoteka je jedan od uzroka veoma niske produktivnosti u razvoju IS. Čak i kada postoje svi podaci koji se u nekoj aplikaciji zahtevaju, zadovoljenje i nekog jednostavnog informacionog zahteva iziskuje značajne programerske napore.

Pasivan odnos korisnika. Svoje zahteve korisnik može da realizuje samo posredstvom sistem analitičara, projektanta i programera, pa se samim tim, zbog njihove niske produktivnosti, zahtevi korisnika sporo i nepotpuno zadovoljavaju.

Sistem za upravljanje bazom podataka je jedan složeni softverski sistem koji treba da omogući:

skladištenje podataka sa minimumom redundanse, korišćenje zajedničkih podataka od strane svih ovlašćenih korisnika, logičku i fizičku nezavisnost programa od podataka, jednostavno komuniciranje sa bazom podataka preko jezika bliskih korisniku.

Baza podataka je kolekcija medjusobno povezanih podataka, uskladištenih sa minimumom redundanse, koje koriste, zajednički, svi procesi obrade u sistemu.

3. Pojam metodologije projektovanja IS-a.

Metodologija je definisani proces razvoja IS, gde se u različitim fazama, na različit način, primenjuju definisani modeli. Osnovni koraci u metodologiji se nazivaju "Model životnog ciklusa" metodologije.

Vrste "modela životnog ciklusa": ***Transformacioni*** (sistem se projektuje po fazama), ***Inkrementalni***

(
svakoj narednoj fazi dodaju se novi detalji o sistemu) i
Kombinovani
(t
ransformaciono se razvijaju manji moduli sistema, a do potpunog modela sistema dolazi se inkrementalno).

4. Klasicni (konvencionalni) pristup.

MODELI: sturkturna sistemska analiza, model objekti-veze, model aplikacije

METODOLOGIJA - Transformacioni životni ciklus: analiza sistema (SSA), specifikacija sistema (MOV), logičko i fizičko projektovanje baza podataka i projektovanje programa, implementacija baze podataka, uvođenje, održavanje.

5. SSA: pojam, osnovni elementi.

Strukturna sistemska analiza (SSA) je metodologija za specifikaciju informacionog sistema. O
na
mo
ž
e
biti
polazna
osnova
za
projektovanje
logi
č
ke
strukture
baze
podataka
metodom
normalizacije
.

Osnovni koncepti za specifikaciju IS u SSA su funkcije, odnosno procesi obrade podataka, tok
ovi
podataka
,
skladi
š
ta
podataka
i
interfejsi
.
Njihov
me
đ
usobni
odnos
se
prikazuje
preko
DTP
.
Grafi

č
ki
simboli
DTP
.

Jednu potpunu specifikaciju IS čine: hijerarhijski organizovan skup dijagrama toka podataka, r
e
č
nik
podataka
koji
opisuje
sadr
ž
aj
i
strukturu
svih
tokova
i
skladi
š
ta
podataka
, s
pecifikacija
logike
primitivnih
procesa
;

6. Dijagrami tokova podataka.

Dijagram toka podatka (DTP) predstavlja model sistema koji sadrži četiri osnovne komponente : procese obrade podataka, objekte sa kojima sistem komunicira, skladišta podataka i tokove podataka koji povezuju ostale komponente sistema u celinu.

Tok podataka mora da ima izvor i ponor. Bilo koja komponeneta DTP može da bude izvor ili ponor. Tokovima se ne mogu neposredno povezati dva skladišta, dva interfejsa, ili skladište i interfejs.

Svako skladište takođe treba da ima barem jedan ulazni i barem jedan izlazni tok, ali dozvoljava

se i da skladište nema ulazni tok, podrazumevajući da se formira i ažurira u nekom drugom sistemu.

Takodje se može vršiti dekompozicija tokova podataka po hijerarhijskim nivoima.

Procesi koji se dalje ne dekomponuju se nazivaju primitivni procesi.

Pored procesa, mogu se dekomponovati i tokovi i skladišta. Dekompozicija tokova i skladišta se ne prikazuje na DTP-u, već u Rečniku podataka, pomoću sintakse za opis strukture podatka.

Najznačajnije pravilo koje se mora poštovati pri dekompoziciji procesa je pravilo balansa tokova: Ulazni i izlazni tokovi na celokunom DTP-u koji je dobijen dekompozicijom nekog procesa P moraju biti ekvivalentni sa ulaznim i izlaznim tokovima toga procesa P na dijagramu višeg nivoa.

7. Rečnik podataka. Rečnik podataka daje opis strukture i sadržaja svih tokova i skladišta

podataka.

Polje je elementarna struktura koja se dalje ne dekomponuje i koja ima svoju vrednost. Polja svoje vrednosti uzimaju domena koji mogu biti: **predefinisani**, **semantički** (kada se definišu posebno). Ograničenja mogu biti prosta (IN, NOT NULL) i složena (prostih ili drugih složenih, vezujići ih logičkim operatorima).

Struktura tokova podataka i skladišta predstavlja neku kompoziciju polja i može biti:

- Agregacija komponenti, “špicaste” zagrade <a,b,c.>.

- Eksluzivna specijalizacija, “uglaste” zagrade [a,b,c], pojavljuje se ili a ili b ili c, ako je [a] - javlja se ili ne.

- Neeksluzivna specijalizacija, “kose” zagrade /a,b,c/, bilo samo jedna, bilo dve, bilo sve.

- Skup komponenti, “vitičaste” zagrade {a}, više puta.

8. Predstavljanje procesa.

Fizički i logički modeli procesa. Logički - specifikacija IS, fizički – implementaciju zadate speci fikacije
.

Osnovni cilj metode SSA može se ostvariti bilo "direktnim modeliranjem", bilo "
snimanjem
".

Direktno modeliranje - dobrom poznavanju sistema. Osnovni koraci kod direktnog modeliranja sistema:

identifikacija osnovnog cilja posmatranog sistema, identifikacija osnovnih procesa sistema, identifikacija neophodnih informacija, identifikacija procesa održavanja.

9. Model entiteta i poveznika (MOV model).

MODEL PODATAKA. Da bi se podaci, koji treba da uđu u bazu podataka, sredili i sistematizovano prikazali koristi se model podataka

.

Osnovni elementi modela podataka: Entiteti - poseduju neke osobine i obeležja, **Atributi** – obeležja.

Za realizaciju baze podataka moramo prikazati i **veze (relacije) između entiteta**: **1:1, 1 : M, M : M.**

Najpoznatiji način za prikazivanje modela podataka je **E-R-A model**
(Entity-Relationship-Atribute) ili kod nas
MOV
(Model Objekti Veze obeležja).

Jak entitet ima osobinu koja omogućava njegovo jednoznačno identifikovanje - primarni ključ entiteta.

Slabi objekat - nasleđuje ključ objekta s kojim je povezan.

Podtip je entitet koji nasleduje primarni ključ jakog entiteta s kojim je povezan.

Agregacija - entitet koji poseduje složeni ključ.

10. UML standard.

Pogled slučajeva korišćenja(Use-case view). Predstavlja se preko statičkog aspekta.

Projektni pogled (Design view). Statički aspekt sistema se ovde prikazuje preko dijagrami klasa i dijagrami objekata, a

dinamički

aspekt preko

dijagrama interakcije.

Procesni pogled (Process view). Prikazuje "niti upravljanja". Preko ovoga pogleda prvenstveno se analiziraju performanse.

Implementacioni pogled (Implementation view). Statički aspekt se prikazuje dijagramima komponenti, a za dinamički se koriste dijagrami interakcije.

Pogled razmeštaja (Deployment view) - prikazuje sistemsko-hardversku topologiju.

SPOMINJU SE: Dijagrami slučajeva korišćenja, aktivnosti, promene stanja, klasa, sekvenci, kolaboracioni dijagrami, dijagrami komponenti i dijagrami razmeštaja.

11. Pojam klase, objekta, atributa, metoda.

Objekat u OO pristupima predstavlja entitet koji je sposoban da čuva svoja stanja.

Klasa pretstavlja skup objekata sličnih karakteristika.

Atributom se podrazumeva osobina klase koja opisuje neku vrednost.

Operacije i metode. Dinamika sistema u OO pristupima se ostvaruje na taj način što jedan objekat šalje drugom događaj. Svi objekti u nekoj klasi imaju iste operacije. Osobina da

objekat poziva neku operaciju drugog objekta ne znajući kojoj klasi ovaj objekat pripada naziva se polimorfizam.